

MICROHOBBY

AMSTRAD

REVISTA INDEPENDIENTE PARA USUARIOS DE ORDENADORES AMSTRAD

Semanal

AÑO I N.º 3

150 Ptas.

Canarias 160 ptas.

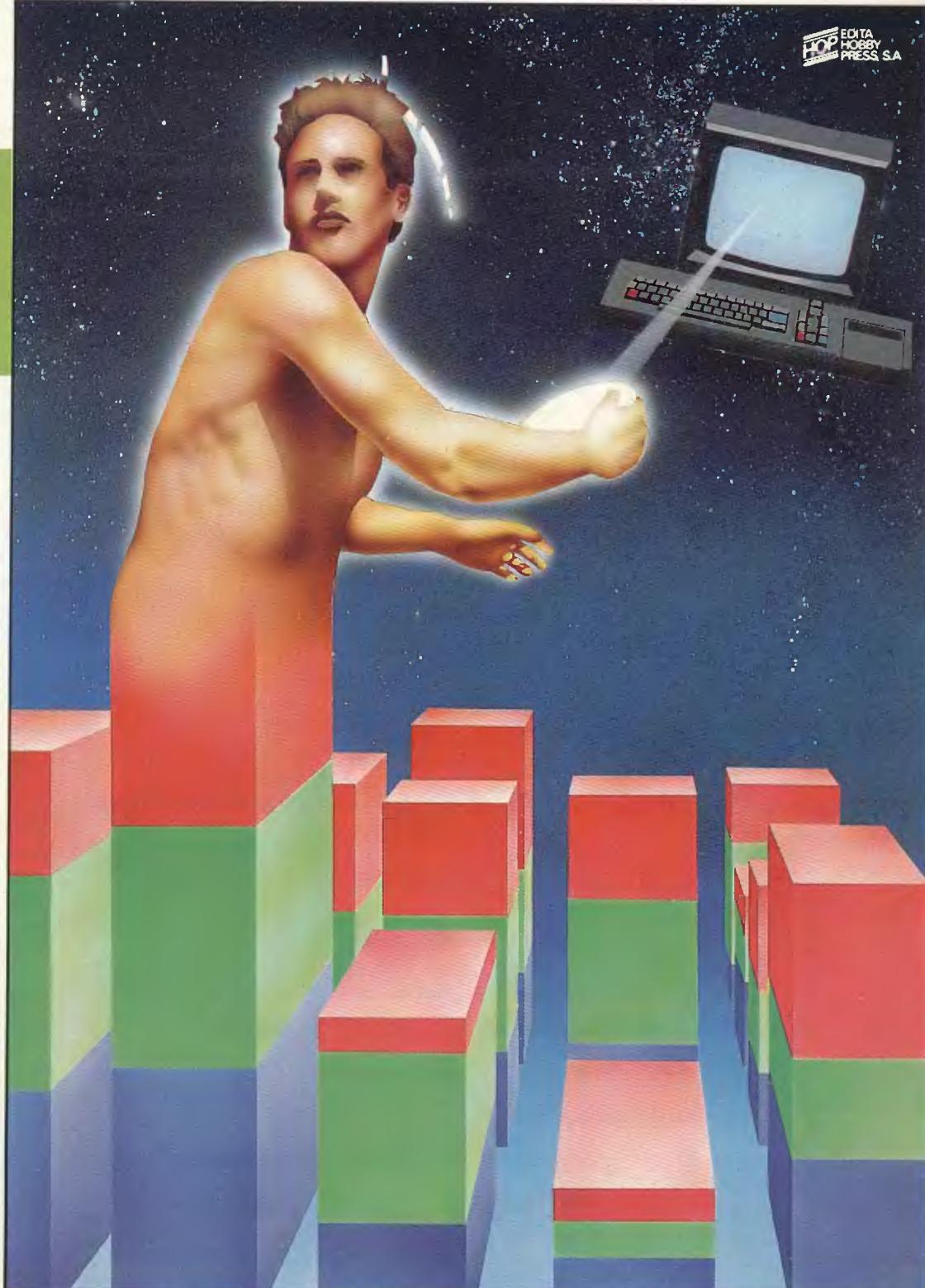
**BIORRITMOS,
ANTICIPA TUS
CICLOS VITALES
DE LOS PROXIMOS
MESES**

**ANALIZAMOS
A FONDO EL
AMSTRAD
128K**

**Análisis:
Decimales
Encolumnados**

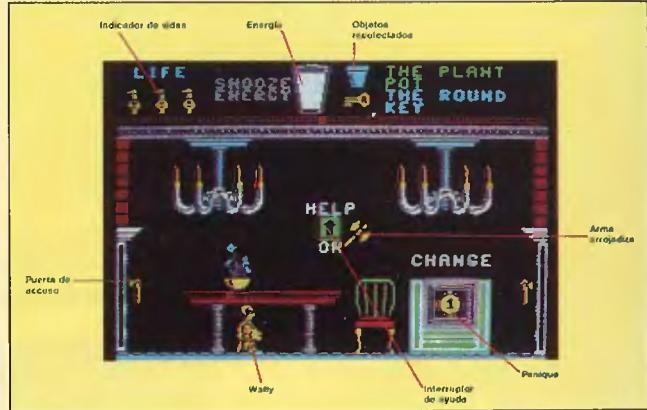
SOFTWARE

**Alien 8:
Una Odisea
en el Espacio**



MICROMANIA.

Sólo para adictos



MICRO Mania

LOTE HOBBY PRESS S.A.

Año 1 N° 4

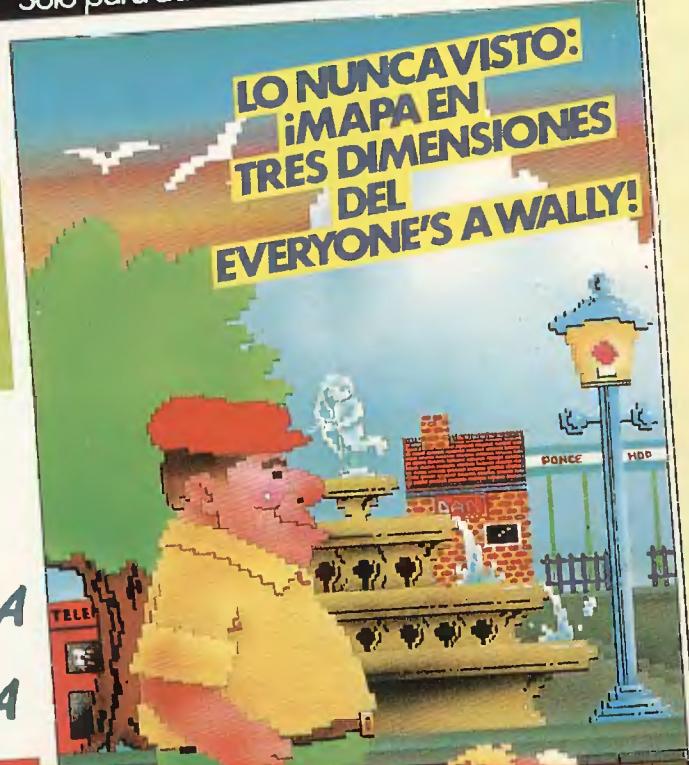
250 Ptas.

MUSICA MAESTRO!



TODA
UNA ORQUESTA
EN TU
AMSTRAD

PATAS ARRIBA
LOS "POKES"
DE
PYJAMARAMA
Y AUTOMANIA



HOBBY PRESS, S.A. Editamos para gente inquieta.

MICR

Franquicia
Postal



HOBBY PRESS, S.A.

Apartado de Correos

n.º 54.062 (Apartados Altos)

MADRID

¡SUSCRIBETE A Microhobby AMSTRAD Y AHORRA 1.600 PTAS

MICRO *Manía*

PODÉTENOS
HOBBY PRESS SA

Año I · N.º 4

Sólo para adictos

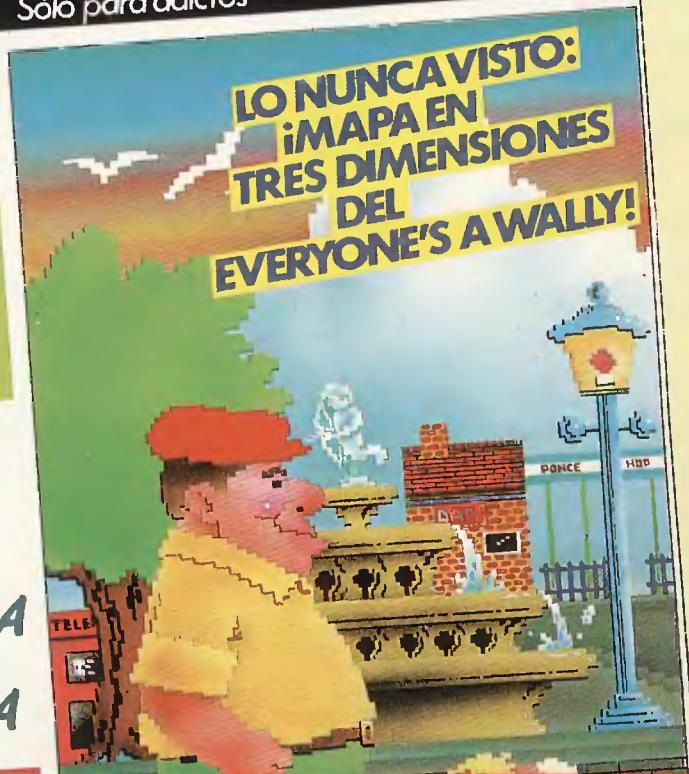
250 Ptas.

¡MUSICA MAESTRO!

**TODA
UNA ORQUESTA
EN TU
AMSTRAD**



**PATAS ARRIBA
LOS "POKES"
DE
PYJAMARAMA
y AUTOMANIA**



HOBBY PRESS, S.A. Editamos para gente inquieta.

TARJETA DE SUSCRIPCION AHORRO/REGALO

(Si lo prefieres suscríbete por teléfono (91) 733 50 12 (91) 733 50 16)

Oferta especial para recibir en tu domicilio todo un año la revista semanal Microhobby AMSTRAD con un descuento de 1.600 ptas., y el regalo de una cinta original por valor de 2.100 ptas.!

FECHA LIMITE DE RESPUESTA:
30 DE NOVIEMBRE DE 1985
¡RESPONDE HOY MISMO!

ISEÑALA EL NOMBRE DE LA CINTA QUE
PREFIERES Y LA RECIBIRAS EN TU CASA,
GRATIS, A VUELTA DE CORREO!

Deseo suscribirme a **Microhobby AMSTRAD** durante un año por sólo **5.900 ptas.**, lo que me supone un ahorro de **1.600 ptas.**
El primer número que deseo recibir es el _____

Envíenme **GRATIS** la cinta de programas que le indico con una (X)

Beach Head

Pole Position

D.T. Decathlon

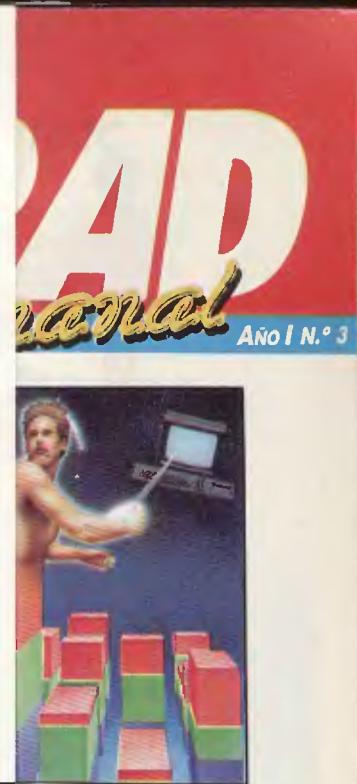
NOMBRE _____ EDAD _____
APELLIDOS _____
DOMICILIO _____
CIUDAD _____ PROVINCIA _____
C. POSTAL _____ TELÉFONO _____ PROFESIÓN _____

Marco con una (X) en el casillero correspondiente la forma de pago que más me conviene.

Talón bancario adjunto a nombre HOBBY PRESS, S. A. Giro Postal N.º _____ Contra reembolso del primer número, junto a la cassette regalo. VISA N.º _____ MASTER CHARGE N.º _____

Fecha de caducidad de la tarjeta _____

Firma: _____



Presidente

Maria Andriño

Consejero Delegado

José I. Gómez-Centurión

Jefe de Publicidad

Concha Gutiérrez

Secretaría de Dirección

Marisa Cogorro

Suscripciones

M.º Rosa González

M.º del Mar Calzada

Redacción, Administración y Publicidad

La Granja, s/n
Polígono Industrial de
Alcobendas
Tel.: 654 32 11

Dto. Circulación

Carlos Peropadre

Distribución

Coedis, S. A. Valencia, 245
Barcelona

Imprime

ROTEDIC, S. A. Ctra. de Irún.
Km. 12,450 (MADRID)

Fotocomposición

Novocomp, S.A.
Nicolás Morales, 38-40

Fotomecánica

GROF
Ezequiel Solano, 16
Depósito Legal:
M-28468-1985

Derechos exclusivos de la revista **COMPUTING with the AMSTRAD**

Representante para Argentina, Chile,
Uruguay y Paraguay, Cia.
Americana de Ediciones, S.R.L. Sud
América 1.532. Tel.: 21 24 64, 1209
BUENOS AIRES (Argentina).

M. H. AMSTRAD no se hace
necesariamente solidaria de las
opiniones vertidas por sus
colaboradores en los artículos
firmados. Reservados todos los
derechos.

Se solicitará control OJD

con microprocesador de ocho bits.

10 Mr. Joystick

El superrobot ALIEN 8, serie 43567ADF, del AÑO 2.256 PC, es el guardián custodio de la nave Amstroblaster, que transporta a través del universo galáctico a los últimos habitantes del planeta Undermunde, en estado de hibernación.

Beach Head

La inexpugnable fortaleza de Kuhn-Lin, es el objetivo de una poderosa flota de invasión que, a través de la armada japonesa, debe desembarcar sus tanques en las arenas de la playa.



14 Primeros pasos

No tiene mucho sentido hablar de programación sin hacer y ejecutar programas.

Por ello, vamos a aumentar los conocimientos de programación que ya poseímos, incluyendo el concepto de variable de cadena y aprendiendo a emplear el **AMSTRAD** como una potente calculadora.

18 Serie oro

En el ciclo biológico de la actividad humana, se distinguen tres funciones principales: física, intelectual y emocional. Cada una de ellas ocupa una parte muy característica en el comportamiento de cada individuo.

Conocer el nivel que vamos a alcanzar en cada uno de ellos, puede ser de gran importancia.

BIORRITMOS te permite determinar con anterioridad de meses estos niveles, lo que te ayudará a programar tus actividades.

Código Máquina 24

Continuamos con nuestra serie de Código Máquina, avanzando un poco más en la ejecución de programas escritos en dicho lenguaje. Se analizan unas cuantas rutinas firmware más, e introducimos el concepto y uso de los **REGISTROS** como herramienta indispensable para la transferencia de datos entre memorias.

También un regalo especial, cuyo nombre es **HEXER** y nos será muy útil en el futuro.

31 Análisis

Ordenar números decimales de forma que su presentación visual sea clara y agradable a la vista, no representa un gran problema para los programadores que utilizan adecuadamente los comandos **MID\$** y **LEN**.

Con ellos conseguimos ordenar decimales de forma que los puntos indicadores de la cifra decimal queden colocados en la misma columna.

Director Editorial
José I. Gómez-Centurión
Director Ejecutivo
Víctor Prieto
Subdirector
José María Díaz
Redactora Jefe
Marta García
Diseño
José Flores
Colaboradores
Francisco Portalo
Pedro Sudón
Miguel Sepúlveda
Francisco Martín
Jesús Alonso
Pedro S. Pérez
Secretaria Redacción
Carmen Santamaría
Fotografía
Carlos Candel
Javier Martínez
Portada
Manuel Barco
Ilustradores
J. Igual, J. Pons, F. L. Frontán,
J. Sepien, Pejo, J. J. Mora
Edita
HOBBY PRESS S.A.

Presidente
María Andriño
Consejero Delegado
José I. Gómez-Centurión

Jefe de Publicidad
Concha Gutiérrez

Secretaría de Dirección
Marisa Cogorro

Suscripciones
M.º Rosa González
M.º del Mar Calzada

Redacción,
Administración
y Publicidad
La Granja, s/n
Polígono Industrial de
Alcobendas
Tel.: 654 32 11

Dto. Circulación
Carlos Peropadre

Distribución
Coedis, S. A. Valencia, 245
Barcelona

Imprime
ROTEDIC, S. A. Ctra. de Irún.
Km. 12,450 (MADRID)
Fotocomposición
Novocomp, S.A.
Nicolás Morales, 38-40
Fotomecánica
GROF
Ezequiel Solana, 16
Depósito Legal:
M-28468-1985

Derechos exclusivos
de la revista
COMPUTING with
the AMSTRAD

Representante para Argentina, Chile,
Uruguay y Paraguay, Cia.
América de Ediciones, S.R.L. Sud
América 1.532. Tel.: 21 24 64. 1209
BUENOS AIRES (Argentina).

M. H. AMSTRAD no se hace
necesariamente solidaria de las
opiniones vertidas por sus
colaboradores en los artículos
firmados. Reservados todos los
derechos.

Se solicitará control OJD

MICROHOBBY

AMSTRAD

REVISTA INDEPENDIENTE PARA USUARIOS DE ORDENADORES AMSTRAD

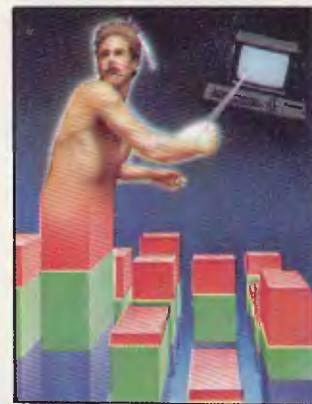
Semanal

AÑO I N.º 3

Año I • Número 3 • 17 al 23 de Septiembre de 1985
150 ptas. (sobretasa Canarias, 10 ptas.)

6 Banco de pruebas

El CPC-6128 llega a España, banco de pruebas con motivo de su presentación oficial, analiza a fondo el nuevo modelo de **AMSTRAD**. Con más memoria, nuevo diseño, dimensiones reducidas respecto a los anteriores modelos, e incorporando el sistema operativo CP/M PLUS; el más potente en el mundo de los ordenadores con microprocesador de ocho bits.



18 Serie oro

En el ciclo biológico de la actividad humana, se distinguen tres funciones principales: física, intelectual y emocional. Cada una de ellas ocupa una parte muy característica en el comportamiento de cada individuo.

Conocer el nivel que vamos a alcanzar en cada uno de ellos, puede ser de gran importancia.

BIORRITMOS te permite determinar con anterioridad de meses estos niveles, lo que te ayudará a programar tus actividades.

10 Mr. Joystick

El superrobot ALIEN 8, serie 43567ADF, del AÑO 2.256 PC, es el guardián custodio de la nave Amstroblaster, que transporta a través del universo galáctico a los últimos habitantes del planeta Undermunde, en estado de hibernación.

Beach Head

La inexpugnable fortaleza de Kuhn-Lin, es el objetivo de una poderosa flota de invasión que, a través de la armada japonesa, debe desembarcar sus tanques en las arenas de la playa.



14 Primeros pasos

No tiene mucho sentido hablar de programación sin hacer y ejecutar programas.

Por ello, vamos a aumentar los conocimientos de programación que ya poseímos, incluyendo el concepto de variable de cadena y aprendiendo a emplear el **AMSTRAD** como una potente calculadora.

Código Máquina 24

Continuamos con nuestra serie de Código Máquina, avanzando un poco más en la ejecución de programas escritos en dicho lenguaje. Se analizan unas cuantas rutinas firmware más, e introducimos el concepto y uso de los **REGISTROS** como herramienta indispensable para la transferencia de datos entre memorias.

También un regalo especial, cuyo nombre es **HEXER** y nos será muy útil en el futuro.

31 Análisis

Ordenar números decimales de forma que su presentación visual sea clara y agradable a la vista, no representa un gran problema para los programadores que utilizan adecuadamente los comandos MID\$ y LEN.

Con ellos conseguimos ordenar decimales de forma que los puntos indicadores de la cifra decimal queden colocados en la misma columna.

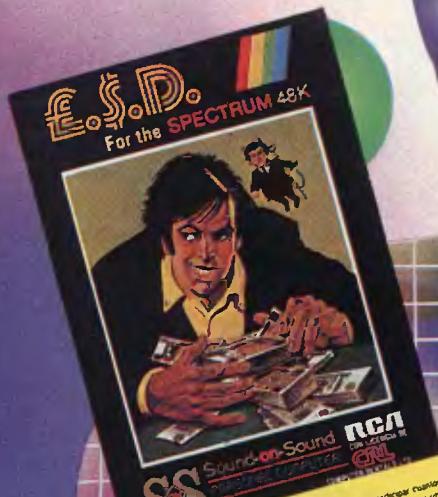
Sound on Sound es una marca registrada
producida y distribuida por **Iberotón, s. a.**
Telf. 671.22.00 / 04/08/12/16

Sound on Sound es una marca registrada
producida y distribuida por **Iberofón, s. a.**
Telf. 671.22.00 / 04/08/12/16

una
stribuida
Telf. 671.22.00

04/08/12/16

Sound

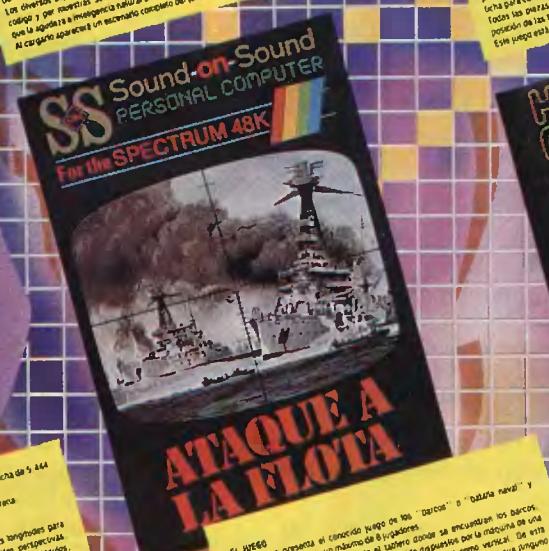


DOMINÓ

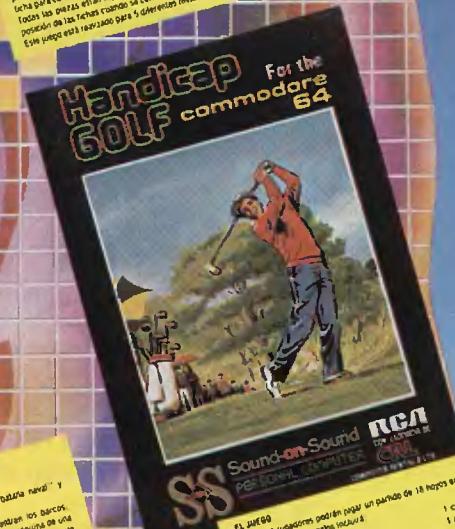


3 en una taza de 5
1 cucharada de azúcar
1 pizca

EL JUEGO
Tú y los jugadores podrán jugar un partido de 18 segundos en una cancha de 9' x 44'.
TIEMPO: 10 partidos por sesión. **TIERRA:** 4 plazas de polvo. **REGLA:** 1 cancha de arena
y 1 grava.
3 puntos de gol.
1 cancha de arena.
Se juega una vuelta de 18 segundos. Inmediatamente que los jugadores toquen la cancha de arena se considera que el juego ha comenzado. Los jugadores deben respetar las perspectivas. Tú, como baterista, tienes que agredir, bolas, maltratar, lugay y controlar. Los balones te representan con una indicación de la distancia en yardas. Los que pasan te caen en la cancha. Una recta con una flecha indica la dirección en la que debes golpear la pelota o la cancha. Esto se indica con una flecha que se convierte en tu vista perspectiva. Cada vez que golpeas la cancha de arena con la vista perspectiva se pierde una parte de la cancha. Se habrá visto "el" cuadro ascendiendo un cerro y desapareciendo de la perspectiva. Ya en plana, se habrá visto "el" cuadro descendiendo un cerro y desapareciendo de la perspectiva. Una vez que se pierda la cancha de arena, ya no se puede volver a golpearla. Deberás dar una aplaudida a tu casa. Una vez que se pierda la cancha de arena, ya no se puede dar una aplaudida a tu casa. Una vez que se pierda la cancha de arena, ya no se puede dar una aplaudida a tu casa.



ATAQUE A LA FLOTA



EL JUEGO
Uno o dos jugadores podrán jugar un partido de 18 horas en una cancha de 3
metros. Tu equipo de pates incluirá:
- 1 portero
- 1 defensa
- 1 centrocampista
- 1 delantero
- 1 mediocampista
- 1 centrocampista defensivo
- 1 centrocampista ofensivo
- 1 delantero central
- 1 delantero exterior
- 1 defensa central
- 1 defensa exterior
- 1 portero suplente
- 1 defensa suplente
- 1 centrocampista suplente
- 1 centrocampista defensivo suplente
- 1 centrocampista ofensivo suplente
- 1 delantero central suplente
- 1 delantero exterior suplente
- 1 defensa central suplente
- 1 defensa exterior suplente
- 1 portero de repuesto
- 1 defensa de repuesto
- 1 centrocampista de repuesto
- 1 centrocampista defensivo de repuesto
- 1 centrocampista ofensivo de repuesto
- 1 delantero central de repuesto
- 1 delantero exterior de repuesto
- 1 defensa central de repuesto
- 1 defensa exterior de repuesto

1 cup oil
1 dozen

Uno o dos días
varían
a pieza de materiales
y gastos de gasoil
de trabajo
Se hará una visita de los 10 hogares, recogiendo las impresiones
de cómo se vive en cada hogar. Se pondrá especial atención a las necesidades
de los hogares más vulnerables. Algunos hogares tienen
expresiones que representan con una indicación de distancia y duración
que el hogar mantiene cada uno.
Al finalizar la visita se hará un resumen de lo visto.
Esta noche se hará una reunión en la iglesia de San Juan.
La reunión es para convocar a todos los vecinos a la sede de la comisión
de apoyo y orientación en el desarrollo social de la población.
Se invita a los vecinos a que asistan a la reunión.
Se invita a los vecinos a que asistan a la reunión.
Se invita a los vecinos a que asistan a la reunión.

IMPRESORA 130 CPS

Sale al mercado una nueva impresora para **Amstrad**, compatible con IBM.

La impresora es comercializada en España por la firma **INDESCOMP**. Este nuevo periférico, permite su utilización con toda la gama de ordenadores **Amstrad**, bien sea con los CPC-464, CPC-664 o CPC-6128. Además de poder utilizarse con casi todos los ordenadores del mercado; al admitir tanto el interface CENTRONICS, usado por los **AMSTRAD**, como el RS232C.

Una característica definitiva de este modelo, es su velocidad; 130 caracteres por segundo.

El diseño del periférico es agradable, con unas dimensiones de 38.40x31.50x12.50 cm lo que hace que ocupe un reducido espacio en la mesa de trabajo.

La impresión se realiza por matriz de puntos con una densidad de 9x9, en caracteres alfanuméricos y de 6x12 en gráficos.

Permite la utilización de 6 tipos de letra:

Normal 80 columnas.

Condensada 132 columnas.

Alargada 40 columnas.

Condensada/

Doble ancho 66 columnas.

Elite 96 columnas.

Elite alargada 48 columnas.

Admite los dos tipos de papel; folios y papel continuo. La anchura del papel a utilizar puede oscilar entre 10,1 cm y 254 cm.

COMPOSITOR MUSICAL PARA AMSTRAD

KUMA COMPUTERS no descansa en su labor de lanzar al mercado más y más software para **AMSTRAD**.

El nuevo producto se llama «**Music Composer**» y es, como su nombre sugiere, una utilidad pensada para permitir componer música, tanto si se trata de usuarios noveles como experimentados.

A medida que vamos componiendo nuestra melodía, aparece en pantalla en forma de notación musical, esto es, usando el pentagrama.

Las composiciones pueden almacenarse en cassette para uso futuro, y existen también ayudas para copiarlas o unir unas con otras.

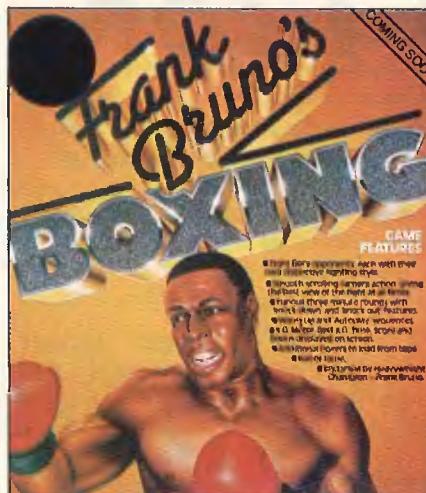
El «**Music Composer**» viene en cassette con manual a un precio de 2.300 pesetas. (en Inglaterra).

Aclaración sobre el Easy Draw:

Todos aquellos usuarios que deseen emplear el Easy Draw con el CPC 464 sin unidad de disco por favor borren del programa las líneas 170 y 180.

Rogamos disculpas por las posibles molestias.

Primera plana



FRANK BRUNO'S BOXING

Elite, casa con justificada fama en el mundo de los juegos para ordenador, abre su camino en el **AMSTRAD** con un espectacular programa de boxeo.

Frank Bruno's Boxing, enfrenta a nuestro morenito héroe con los más inesperados rivales.

El camino hasta conseguir el campeonato mundial es largo y requiere un gran esfuerzo, cada nuevo adversario es más difícil de vencer, sus golpes son más contundentes y su técnica más depurada.

La mayor innovación del juego consiste en el número de rivales contra los que combatimos. Dada la escasez física de memoria, se ha recurrido a un truco para solucionar este problema; los distintos adversarios se encuentran en la segunda cara de la cinta.

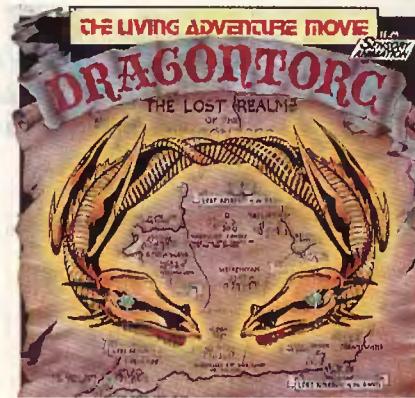
La carga de un nuevo contrincante se realiza introduciendo el código que hemos obtenido al vencer al anterior, con lo cual se mantiene viva la incertidumbre de saber cuál será el próximo rival y nos obliga a vencer a los que están antes. Cada contrincante utiliza sus golpes secretos y debe ser vencido por K.O. antes de tres minutos.

La victoria está en tus puños.

DRAGONTORG

Hombres lobos en Wolfwood, fantasmas la cripta maldita, enanos en Dreamdown, murciélagos en las cavernas de la puerta del Infierno, son algunas de las criaturas y lugares que visitarás en el nuevo juego de **HEWSON CONSULTANTS**.

Tus hechizos y el mago MERLYN, facilitarán el peregrinaje de MAROC, a través del reino de CANTI.



3-D VOICE CHESS

El ajedrez, todo un clásico en los programas para ordenador, adquiere una nueva dimensión en el producto realizado por **DEEP THOUGHT SOFTWARE**.

Este programa compatible con el CPC-464 y el CPC-664, incorpora dos nuevos conceptos en los juegos de ajedrez:

La pantalla ofrece una visión en tres dimensiones del tablero y las fichas; reproduciendo el ángulo visual que tendría un jugador en una partida real.

Todos los movimientos son acompañados por la voz del ordenador, que nos repite la jugada realizada.

Otras características:

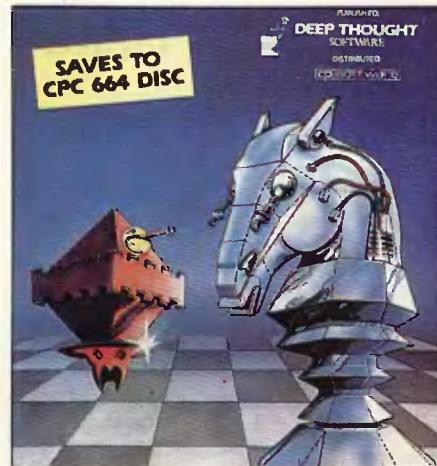
Siete niveles de juego.

Opción para gravar el programa en disco.

Librería de aperturas.

Ánalisis de partidas.

Menú de ayuda en las situaciones desesperadas.



ANALIZAMOS A FONDO EL AMSTRAD 128K

Ya está aquí el nuevo AMSTRAD, con nada menos que 128 Kbytes de memoria RAM. Más pequeño, más manejable, y mucho más potente.

No cabe duda que más memoria es sinónimo de más potencia.

Todos los fabricantes de software son conscientes de ello, por lo que se tiende en la actualidad a crear programas que hacen un uso y consumo masivo de la memoria.

Sin embargo, hasta ahora las casas de software se veían limitadas por partida doble: los fabricantes de hardware, al menos a nivel de pequeños equipos, no pasaban del límite de las 64 K, y el público, quien más, quien menos, se conformaba con lo que había: CP/M reinaba en solitario, y hablar o pensar siquiera en algo distinto de lo establecido era un delito de lesa majestad.

Afortunadamente para todo el mundo, hace unos pocos años, los componentes de los ordenadores, sobre todo las memorias RAM, se abarataron tanto gracias al avance de la tecnología, que pensar en computadores con grandes cantidades de memoria RAM disponible dejó de ser una quimera: se convirtió en el mayor argumento de marketing para cualquiera que tuviera dos dedos de frente.

Cuestión de lógica

IBM, sin duda, tenía al menos un dedo de frente, porque en 1981 lanzó un ordenador de 16 bits, volviendo del revés el concepto de microinformática y acuñando la moneda llamada «ordenador personal», una de las mayores gallinas de los huevos de oro inventados hasta la fecha.

Toda buena idea engendra imitadores, así que pronto apareció en el mercado toda una constelación de ordenadores compatibles con el IBM «Personal Computer».

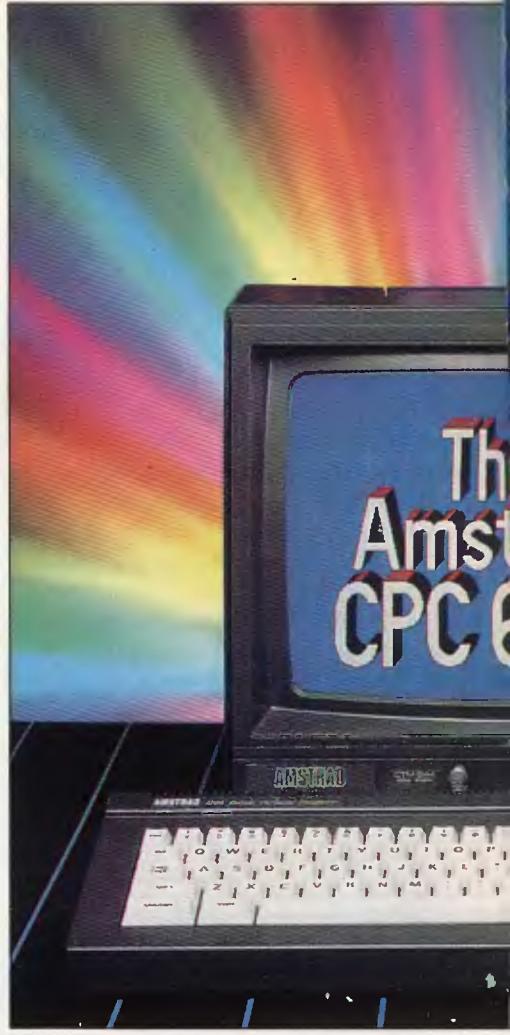
Ahora bien, ¿Cómo luchar contra el nombre mágico del gigante azul? Los que tenían el segundo dedo de frente encontraron la respuesta justa sorprendentemente rápido: más memoria que IBM al mismo precio.

Los ordenadores de 128 Kbytes acababan de nacer para el gran público.

Bien, ya puestos, ¿por qué no intentar hacer (es decir, vender) ordenadores de 128 Kbytes de RAM pero con procesadores de 8 bits (o sea, más baratos), como por ejemplo el Z80?

Total, sólo hay que modificar el sistema operativo CP/M, adaptándolo para manipular 128 K, y algún que otro programa para que corra bajo el nuevo sistema, con el aliciente de que la compatibilidad con versiones CP/M anteriores (*la 2.2*) se mantenga.

Dicho y hecho: máquinas basadas en el Z80, ejecutando el flamante CP/M 3.1, dedicadas a la gestión, reclamaron su derecho a la vida a golpe de anuncio.



La renovación de los 8 bits

Poco a poco, esta nueva concepción de un ordenador de 8 bits se infiltró silenciosamente en la mente de los fabricantes de «home computers», pero su lanzamiento se retrasó hasta «el momento oportuno», es decir, hasta que los actuales modelos dejaran de venderse o la competencia fuera demasiado feroz. El día D ha llegado. En este año se han dado todas juntas las circunstancias ideales para la aparición de estos micros en nuestros hogares: caída de precios, presión del usuario, software adecuado, etc.

Casi todas las marcas consagradas han tomado su lugar en la salida de la carrera de los 128 K y... Amstrad quiere ganarla.

Por ello ha nacido el CPC6128, el intento más serio visto hasta ahora en nuestro mercado de vender un ordenador de gestión a un precio cercano a uno de juegos.

En este sentido, el CPC6128 mantiene la misma línea que sus antecesores, inclinándose progresivamente al mundo de las «aplicaciones se-



El nuevo traje del Amstrad

Y a todo esto, ¿cómo es el CPC-6128?

Bueno, pues la verdad es que es bonito. Se ha vuelto a cambiar el diseño: el ordenador es bastante más pequeño que el 664, y más plano.

El teclado posee un atractivo color gris pastel, y tiene la misma calidad que en los modelos anteriores, sólo que su tacto es mucho más agradable y suave, y la respuesta inmediata.

Todas las teclas son del mismo color, y se ha suprimido la división del teclado en dos bloques. Aunque es indudable que así se consigue un ahorro de espacio importante, la desgraciada proximidad de la tecla RETURN al bloque numérico puede causar errores de tecleo con facilidad. De cualquier forma, esto es sólo un pequeña incomodidad, ya que existe otra tecla etiquetada como ENTER que cumple la misma función, en principio, y cuyo acceso está mucho más logrado. Algunas teclas, como la de control, han sido colocadas en otro lugar del teclado, con idea de facilitar su acceso. Desaparece el bloque con las cuatro teclas del cursor y la tecla COPY en medio. Dichas teclas se ubican ahora en el bloque numérico, adoptando una disposición y un aspecto un tanto diferentes. La tecla COPY se coloca ahora en la parte inferior izquierda del teclado.

En resumen, muy bien el teclado: suave al tacto, agradable a la vista y bien diseñado.

Se echa en falta la división en dos bloques, aunque gracias a ello la ganancia de espacio ha permitido disminuir las dimensiones del ordenador.

El CPC6128 está también dotado para el intercambio de información con el exterior como el 664. Posee:

- Conexión para joystick
- Conexión para cassette
- Salida Centronics
- Interface para una segunda unidad de disco
- Salida estéreo
- Port de expansión

Lo más interesante, desde el punto de vista de las comunicaciones, tal vez sea la salida Centronics, que permite emplear docenas de impresoras y otros periféricos, y el port de expansión, que capacita al CPC6128 para beneficiarse de programas escritos en ROM (como las RSX) y, quién sabe, tal vez de más memoria en el futuro.

Banco de pruebas

Se echa de menos una interface serie RS232 y modems, aunque sabemos que están disponibles en Inglaterra.

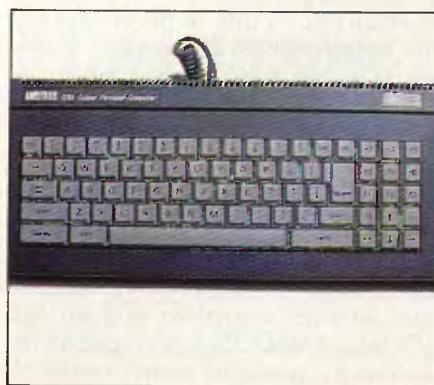
En fin, el CPC6128 se encuentra bastante bien dotado para las comunicaciones, aunque las distancias que separa nuestro propio servicio de comunicaciones con respecto al de otros países europeos, impide prácticamente el uso de modems para conectarse a otros ordenadores o bases de datos públicas.

Memoria para dar y tomar

Hoy en día casi todo el mundo interesado en la informática sabe que el Z80 es el microprocesador más popular dentro del mundo de los 8 bits. También sabe que sólo puede direccionar 64 K de memoria, así que ¿dónde está el truco?, porque hemos dicho a los cuatro vientos que el CPC6128 tiene 128 K.

Realmente, esta pregunta se contesta a medias en el caso del 664, ya que este ordenador tiene en total 96 K de memoria, 32 de ROM, 64 de RAM. En cualquier caso, más de 64.

El asunto está en que, mediante determinadas técnicas de hardware controladas por programa, se puede engañar al microprocesador haciéndole creer que sólo accede a 64 K, pero teniendo realmente muchas más. A grosso modo, basta con alterar unos cuantos bits y donde antes había RAM, ahora el Z80 lee de la ROM. Es decir, ambas memorias solapan.



COMANDOS PARA EL MANEJO DEL SEGUNDO BLOQUE DE MEMORIA

MANEJO DE GRAFICOS

Comando	Parámetros	Función
I SCREENCOPY	3 parámetros: sección de pantalla, pantalla destino, pantalla origen.	Copia el contenido de la pantalla destino en la pantalla origen.
I SCREENSWAP	3 parámetros: sección de pantalla, número de pantalla, número de pantalla.	Intercambia el contenido de dos pantallas.

Nota: la palabra «pantalla» se refiere a un bloque de memoria de 16 k, que puede estar en los 64 kbytes extra o referirse a la pantalla «normal» de gráficos y/o texto.

ALMACENAMIENTO DE DATOS

Comando	Parámetros	Función
I BANKOPEN	1 parámetro: longitud en caracteres que contendrá cada récord.	Divide la memoria extra en récords de acceso aleatorio.
I BANKWRITE	3 parámetros: variable entera, cadena a almacenar, número de récord.	Almacena en memoria una cadena en el récord en uso. La variable entera se usa para devolver información acerca de la operación. El número de récord es opcional. Si no se especifica, se escribe en el de uso corriente y el puntero se incrementa en una unidad para apuntar al siguiente.
I BANKREAD	3 parámetros: variable entera, variable de cadena, número de récord.	Lee del récord correspondiente y devuelve lo leído en el segundo parámetro. El tercer parámetro es opcional.
I BANKFIND	3 parámetros: variable entera, cadena a buscar, número de récord.	Busca una cadena a lo largo de todos los récords. El número de récords es opcional. Devuelve información acerca del éxito o fracaso de la búsqueda en la variable entera.

En el caso del CPC6128, la memoria ROM asciende a 48 K, esto es, el Z80 tiene que bregar con un total de 176 Kbytes de memoria. Para ello, se emplea la misma técnica con la ROM que la descrita anteriormente, y además, para las 64 K RAM extra, se recurre a lo que se denomina como «conmutación de bancos», es decir, también muy a groso modo, el Z80 accede a toda la memoria disponible pero por bloques de 64 K, o bien, en el caso concreto del CPC6128, se intercambian bloques de 16 K.

Estas dos acciones se toman dependiendo fundamentalmente de qué sistema operativo esté en uso (CP/M o AMSDOS) y para qué se requiera la memoria extra: como al-

macen de imágenes de pantalla o datos (AMSDOS) o como área de programa propiamente dicha (CP/M).



Así, el usuario interesado exclusivamente en emplear software comercial para resolver problemas concretos de su negocio, no necesita preocuparse para nada de todo este enredo; el propio programa que se adquiera se encarga de manejar la memoria extra o incluye instrucciones muy claras para llevar a cabo su instalación en el entorno que sea.

128 K accesibles desde BASIC

Sin embargo, pensando sin duda en los aficionados a la programación o al «hágalo usted mismo», Amstrad ha incluido una serie de comandos que permiten manejar desde Basic el segundo banco de memoria.

En aras de la comodidad y la compatibilidad con anteriores modelos, dichos comandos no residen en el intérprete de Basic: se entrega en uno de los dos discos que acompañan al ordenador en forma de dos programas, uno en Basic, probablemente el cargador, y un fichero binario que realiza el trabajo duro.

La nueva extensión del lenguaje permite utilizar la memoria extra para tres propósitos principales:

1. ALMACENAR PANTALLAS GRAFICAS. Podemos tener simultáneamente en memoria CUATRO PANTALLAS DE GRAFICOS COMPLETAS, con la consiguiente ventaja para, por ejemplo, programas de juegos. No es necesario decir que, con un poco de ingenio y de bibliografía, estas pantallas pueden convertirse en muchas más, combinando las posibilidades de almacenamiento del disco con unos cuantos trucos para conseguir que los gráficos ocupen menos memoria (por ejemplo, almacenar tiras de imágenes en lugar de pantallas completas).

2. ALMACENAMIENTO DE DATOS Y VARIABLES. Esta posibilidad apenas necesita explicación: baste decir que se podrán emplear un montón de Kbytes para almacenar datos y variables del programa en curso. Resultado: programas de mucha mayor longitud y potencia.

3. DISCO VIRTUAL. Tal vez la más apasionante de las tres para los interesados en hacerse a medida su propio software de utilidades. Mediante los nuevos comandos, podemos organizar la memoria extra en RECORDS que almacenan datos de forma similar a los ficheros secuenciales de disco, usados por AMSDOS.



La ventaja: el increíble aumento de velocidad en las operaciones de lectura/escritura y de búsqueda de datos.

Esto, combinado con la facilidad de empleo en el **Amstrad CPC6128** del firmware y del lenguaje máquina en general mediante la instrucción Basic CALL, DOTAN A ESTE ORDENADOR DE UNA POTENCIA QUE NINGUN EQUIPO DE ESTE PRECIO POSEE EN NUESTRO MERCADO, en el momento actual.

Salvo por estos comandos, el Basic del CPC6128 es análogo al de CPC664.

CP/M 3.1: la flor y nata

Hemos mencionado antes que con el CPC6128 se entregan dos discos. Contienen el nuevo sistema operativo (CP/M 3.1), una nueva versión del lenguaje LOGO de Digital Research, el sistema operativo anterior (CP/M 2.2), el LOGO para este, y el sistema GSX (*siglas de Graphics System eXtension*), también de Digital Research.

GSX es un sistema gráfico un tanto especial, ya que, más que añadir comandos para el manejo de gráficos al CP/M, lo que hace es capacitar a cualquier programa que cumpla una serie de especificaciones standard, para que él manipule cosas como impresoras y plotters.

Así se remedia en parte uno de los fallos del CP/M, la dificultad en el uso de salida gráfica para algunos tipos de aplicación.

Aquellos que conozcan CP/M 2.2 quedarán agradablemente sorprendidos cuando usen la nueva versión, 3.0.

Al disponer de mucha más memoria, el juego de órdenes CP/M se ha ampliado mucho: algunas se han suprimido o cambiado por otras, y todas se han mejorado.

Los programas en lenguaje ensam-

blador no tendrán motivo de queja: una cara completa de uno de los discos que se entregan con el CPC6128 está dedicada a ellos. Hay de todo: un editor de texto, un macroensamblador, un linkador (editor de enlaces), un programa que crea código máquina relocalizable, etc.

Esperemos que esta vez Digital Research se haya tomado la molestia de que todo sirva para el Z80 en lugar de para el 8080, porque en manuales del CPC6128 no viene una sola palabra acerca de estos programas.

Se mantiene el tradicional grupo de comandos residentes, pero la TPA (*Transient Program Area*), es decir, el área de memoria dedicada a programas transeuntes, no residentes, queda ampliada a un máximo de 61 K. Las aplicaciones que aprovechen esto al máximo, junto con el sistema gráfico GSX, serán algo digno de verse.

Esta versión poco tiene que ver, en filosofía, con las anteriores. Su manejo es mucho más sencillo, y los programas de uso más habitual (*copia de ficheros, formateado de discos, etc.*) ahora se rigen por menús en lugar de por comandos. En la medida de lo posible, se ha tratado de hacer uso de las facilidades del **Amstrad** en cuanto a gráficos y ventanas.

CP/M 3.1 es tan completo y profundo, que sin duda merece un detallado estudio aparte que por razones de espacio no podemos ofrecer en esta primera toma de contacto con el **Amstrad CPC6128**. Trataremos de remediar esta laguna rápidamente.

Logo: aprender jugando

Cuando hablábamos del CPC664, no pudimos por menos que elogiar cumplidamente la versión de LOGO, también de Digital Research, como se merecía. El LOGO que corre bajo CP/M 3.1 es aún mejor. Se han añadido multitud de comandos útiles en todas las áreas que abarca este lenguaje. Con más razón aún que antes, casi todo lo que podemos hacer en **Amstrad** lo podemos realizar en LOG, pero con una gran diferencia que se valora positiva o negativamente según las preferencias de cada uno: LOGO se creó por un grupo de psicólogos e informáticos para enseñar conceptos abstractos, para enseñar a razonar, en definitiva. Es el idioma educativo por excelencia.

Por ello, es un lenguaje altamente



estructurado en procedimientos, no permite las libertades de la sentencia Goto del Basic, y potencia al máximo la creatividad, construyendo los programas de arriba a abajo, de forma totalmente interactiva. Logo, como Basic, también es un intérprete.

No podemos resistir la tentación de recalcar que Logo está mucho mejor preparado que Basic para el tratamiento de palabras.

Digital Research, consecuente con ello, ha dotado al nuevo Logo de todos los comandos necesarios para tal fin, con lo que el desarrollo de programas de Inteligencia Artificial, por ejemplo, es relativamente sencillo, combinando las posibilidades de disco con las 128 K de memoria.

Todos los interesados en este apasionante mundo, campo virgen para la investigación, encontrarán en este lenguaje una inestimable ayuda a bajo precio.

Respecto a los manuales, sólo ha caído en nuestras manos la versión inglesa: es excelente.

Esperemos que dichos manuales traducidos mantengan ese nivel de calidad.

Último aviso

Hace no mucho tiempo, en esta revista, planteábamos la pregunta crucial de «¿qué aporta de nuevo el...?»

Cada vez que un nuevo producto nace, esta pregunta cobra nueva vida, porque resume en muy pocas palabras la duda acerca del éxito o del fracaso de una idea, la filosofía que siempre se esconde tras un nuevo ordenador.

En este sentido, creemos que el CPC6128 contribuye con cosas muy importantes al acervo **AMSTRAD**: más memoria, más potencia y la promesa de un software mucho más sofisticado y amigable, hecha realidad en el sistema operativo CP/M 3.1 y en la magnífica implementación de Logo que le acompaña.

La máquina posee un lugar aún más definido dentro del campo de la pequeña gestión (tal vez no tan pequeña), dejando la puerta abierta, según costumbre de **Amstrad** para el alegre mundo de los juegos por ordenador.

Pronto, quizás ya mismo, el CPC6128 estará con nosotros.

ALIEN 8

COMPATIBLE
CPC 464
CPC 664



En la lejana constelación de ULTIMATEA, los habitantes del planeta UNDERMUNDE, se encuentran en grave peligro de extinción.

Una terrible plaga, el SIDI (Síndrome de Inmuno-Deficiencia Informática), hace estragos en la población que no encuentra remedio alguno para combatirlo.

Los habitantes de UNDERMUNDE, habían conseguido almacenar una gran cantidad de conocimientos e información, llegando a ser la civilización más avanzada de su galaxia. Y ahora su fin estaba próximo.

El gobierno de UNDERMUNDE, reunido en pleno, decidió intentar perpetuar la especie en otra galaxia, enviando una expedición en busca

de algún planeta favorable a sus condiciones de vida.

Después de consultar con los científicos y astrónomos más lúcidos del país, se decidió que el más apto para la vida era KNIGHTCALORE, un planeta del sistema solar que se encuentra a varios millones de años luz.

Como integrantes de la expedición, se seleccionó a los ejemplares más perfectos de la especie, y a los científicos más prestigiosos de los centros de investigación.

Todos ellos fueron hibernados criogénicamente para poder superar el paso del tiempo e introducidos en la nave interestelar, junto con el complejo cibernetico de inteligencia artificial, en el que se almacenaron todos los conocimientos atesorados por esta antigua civilización en el transcurso de su existencia.

El AMSTROBLASTER, nave elegida para recorrer el largo camino, fue entregada al control de un ordenador de ruta y mantenimiento CPC-6128436 54, auxiliado en las labo-

res mecánicas y de infraestructura por el superrobot ALIEN 8, SERIE 43567ADF, construido en el año 2256 PC. de la quinta era undermundiana. Por fin fue lanzada la nave y dio comienzo su larga travesía.

A pesar del transcurso de los sucesivos milenios luz, todos los sistemas funcionan perfectamente, el ordenador central mantiene su ruta y cuida del perfecto estado del soporte vital de los tripulantes criogenizados.

Nuestro robot ALIEN 8, imperturbable por el transcurso de los milenios.



nios, seguía cuidando minuciosamente del mantenimiento de la nave. Pero cuando la nave estaba a 110030 años luz de su destino, el CPC-6128436 54, conocedor de la férrea lealtad del robot, pone en marcha todos los mecanismos ofensivos a su alcance, para evitar que ALIEN 8, cumpla con su misión.

Ante este caos, nuestro hombre ALIEN 8, se mantiene fiel a la misión para la que fue creado en los talleres de la factoría ULTIMATRONICS LTD.; mantener con vida a los argonautas criogenizados.



El CPC-6128436 54, conocedor de la férrea lealtad del robot, pone en marcha todos los mecanismos ofensivos a su alcance, para evitar que ALIEN 8, cumpla con su misión.

Con este objeto el ordenador coloca todo tipo de trampas y obstáculos, para desintegrar a nuestro robot. ALIEN 8, tiene que conseguir reactivar los 24 sistemas de soporte vital, antes de que transcurran los 10000 años luz de que disponemos para reactivar los sistemas vitales de los pasajeros.

Los obstáculos que encontraremos en nuestro camino son:

Ratones mecánicos, minas UXB-64, daleks mutantes, semiesferoides dentados y tetraedros de la muerte.

Nuestros aliados, robodroides controlados a distancia, que son de gran ayuda en la limpieza de zonas minadas.

La forma de mantener la criogenización de los argonautas, es activar ésta con válvulas termáticas, las cuá-



Mr. Joystick

les debemos buscar por toda la nave y llevarlas a las 24 salas de soporte vital.

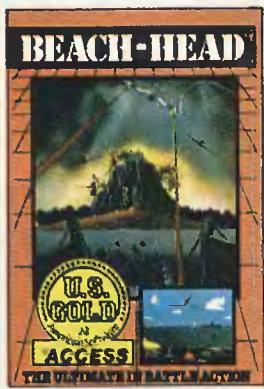
ULTIMATE, con su ALIEN 8 ha seguido fielmente la línea innovadora que marcó con el KNIGHT LORE. La aventura en este caso se ha ambientado en los corredores y estancias de una nave espacial.

Los gráficos son insuperables, el efecto tridimensional asombroso y el movimiento sigue la línea marcada por el KNIGHT LORE. Como sofisticación se han añadido nuevos elementos móviles y nuestro robot, puede dirigir a distancia a los robodroides que se encuentran en algunas salas.

ULTIMATE inició un nuevo camino en la programación con KNIGHT LORE, que continúa ahora con ALIEN 8. **¿QUE VENDRA DESPUES?**



BEACH HEAD



COMPATIBLE

CPC 464
CPC 664

La inexpugnable fortaleza de KUHN-LIN, resiste sin claudicar todos los ataques lanzados por la fuerza invasora.

El camino hasta desembarcar en la playa, está jalónado por difíciles escollos para nuestra flota de desembarco.

Las aguas de la bahía, representan un obstáculo impenetrable, fuertemente custodiadas por la armada nipona, sus portaaviones y acorazados causan grandes bajas en nuestra armada.

Cañones antiaéreos, rechazan las oleadas de caídas enemigas. Después del ataque aéreo, los acorazados y destructores entran en acción; nuestras baterías de largo alcance, intentan hacer blanco en las naves que aparecen en el horizonte.

Las unidades de desembarco llegan a la playa, sobre sus arenas, los tanques comienzan su marcha hacia la sólida fortaleza, los escasos acorazados que llegan a sus pies abren fuego sobre los puntos vitales. Su destrucción es una misión imposible. **BEACH HEAD**, es un juego inspirado en las acciones de desembarco llevadas a cabo en la Segunda Guerra Mundial.

Está realizado en cinco fases, reflejando con exactitud todo el aparato de guerra que representa la toma de una cabeza de playa. La acción comienza en la sala del puente de mando del buque insignia de la flota aliada. El mapa de la zona de invasión, aparece en la pantalla de navegación.

Una vez elegida la ruta que nos llevará a la bahía, damos la orden de avanzar a nuestra flota.

Podemos elegir dos rutas para entrar en la bahía:

Si pasamos directamente por la entrada natural de la bahía, nos encontraremos frontalmente con la armada nipona, que nos espera con sus portaaviones en orden de combate y los caídas en cubierta armados con bombas de penetración.

Manteniendo un duro y prolongado combate con las oleadas de caídas que surgen del horizonte, cuando las distintas escuadrillas cumplido su objetivo, vuelven a la cubierta del portaaviones, éste se retira del campo de batalla y comienza el combate de superficie.

Para evitar el ataque frontal de la armada japonesa, podemos utilizar un estrecho fuertemente defendido por minas y torpedos dirigidos por submarinistas.

Utilizando esta entrada a la bahía, los barcos que logren atravesar el estrecho, atacarán lateralmente a la flota enemiga, evitando que el portaaviones tenga tiempo de lanzar todos sus caídas sobre nuestra flota.

En esta fase del juego, dirigimos el fuego de las baterías antiaéreas sobre los aviones enemigos.

En el combate de superficie, debemos hundir los cuatro acorazados, que desde el horizonte nos bombardean con proyectiles de largo alcance.

Para conseguir bombardear los buques enemigos, hemos de tantear su posición con varios disparos previos, que nos irán indicando en el medidor de precisión de tiro; los metros que nos hemos quedado cortos o los que nos hemos pasado.

Hundidos los acorazados, nuestra flota de desembarco se aproxima a la costa sin oposición alguna. Es el momento de lanzar las divisiones anfibias sobre la playa.

Cada barco que haya superado las fases anterior contiene cuatro tanques, los cuales afrontan la parte final de la misión.

Con ellos debemos atravesar las defensas costeras y destruir los tanques y baterías enemigas, que nos separan de los pies de la fortaleza.

Cada tanque que llega tiene que hacer blanco en los diez puntos vulnerables de ésta, las defensas de



la base dispararán sobre nuestro vehículo, una vez localizada su posición, destruyéndolo a los pocos segundos de colocarse en posición de tiro; lo que nos obliga a disparar con rapidez sobre los blancos.

Una vez destruido nuestro tanque, hemos de conseguir llegar con otro hasta la fortaleza. Repitiendo esta operación varias veces, lograremos alcanzar los diez blancos que necesitamos para destruir las defensas de la fortaleza.

En este momento habremos culminado con éxito la ofensiva sobre la zona costera.

BEACH HEAD, mezcla en el mismo juego varios conceptos diferentes, uniéndo tres juegos de acción arcade; ataque aéreo, combate naval, aproximación a la fortaleza y tiro sobre sus blancos.

El tratamiento gráfico de las distintas fases es bastante acertado, destacando el efecto conseguido en la correspondiente al ataque aéreo; en ella los aviones se nos aproximan desde el horizonte, llegando a situarse sobre nuestras cabezas mientras disparamos sobre ellos con los cañones antiaéreos.

El mismo efecto se repite en el combate de superficie, contra los acorazados, podemos ver como los obuses enemigos caen sobre nosotros y la trayectoria de nuestros disparos, mientras observamos como se eleva el cañón de nuestra batería de fuego.

En la fase de aproximación a la fortaleza, la estructura es totalmente arcade.

Todas las fases están acompañadas de un sonido de explosiones y obuses, bastante real.

Presidido constantemente por la emoción de pasar a la fase siguiente y conseguir ver la fortaleza. Destruirla, requiere un conocimiento más completo del juego y resulta más difícil. No todos los desembarcos tienen éxito.

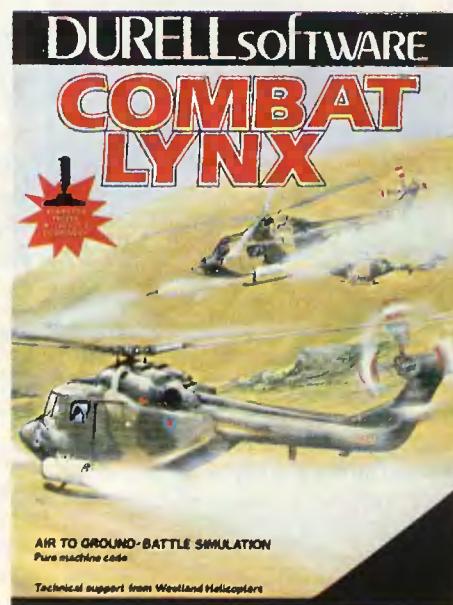
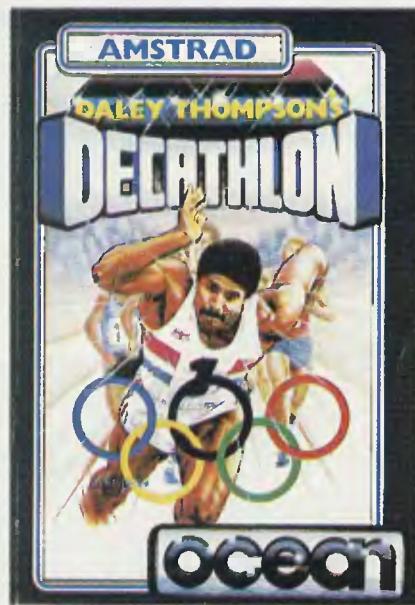


Nuestra dirección:
Castellana, 268, 3.º C
28046 MADRID
Tel.: 733 25 00

círculo de soft

MICROAMIGO S.A.

AMSTRAD



¡Ahora para el AMSTRAD! El super divertido programa de OCEAN sobre las Olimpiadas. 100 m. lisos, disco, jabalina, salto de longitud... y finalmente los 1.500 metros. P.V.P.: 2.100 ptas.

¿Serás capaz de organizar una cabeza de puente en las playas enemigas y tomar sus posiciones? Con este divertidísimo juego puedes intentarlo. P.V.P.: 2.100 ptas.

El simulador de guerra más completo que hayas visto. Atraviesa las líneas del enemigo y destruye sus puntos vitales. Cuentas con radar, mapas, helicópteros, etc. P.V.P.: 2.100 ptas.

¡¡UN INCREÍBLE REGALO POR CADA PROGRAMA!!

Este magnífico reloj digital de cinco funciones puede ser tuyo si pides tus programas al Círculo de Soft. Si tu compra es de dos programas te obsequiaremos con un conmutador TV-Ordenador... y ambos regalos si pides tres programas.



CUPON DE PEDIDO

Recorta o copia este cupón, o pide tus programas por teléfono. Deseo recibir a vuelta de correo el(s) siguiente(s) programa(s).

TITULO _____ P.V.P. _____ ORDENADOR _____

Contrareembolso Giro Postal Talón adjunto a «Microamigo, S. A.
 Tarjeta VISA n.º _____ Fecha caducidad _____

Nombre _____

Apellidos _____

Domicilio _____

Localidad _____ C.P. _____

Provincia _____

Teléfono _____ Edad _____

ESCRIBIENDO NUESTROS PROPIOS PROGRAMAS

Resulta obvio que no tiene sentido hablar de programación sin hacer programas. De ello nos ocupamos en el presente artículo, segundo de la serie «Primeros Pasos», dedicada a todos los recién llegados a la palestra de la programación.



a hemos aprendido a dialogar con nuestro AMSTRAD en el idioma que «él» reconoce: Basic.

También aprendimos como conseguir que el ordenador realizará por nosotros unas cuantas operaciones matemáticas simples, e incluso que escribiera frases en la pantalla, mediante la palabra PRINT, conocida como COMANDO o PALABRA-CLAVE. Por ejemplo:

PRINT 4+4 [ENTER]

(donde [ENTER] quiere decir presionar la tecla marcada ENTER) mostrará en la pantalla el esperado 8, resultado correcto de nuestra operación. Obsérvese la existencia de un espacio entre PRINT y el primer 4. Si intentamos:

PRINT 4+4 [ENTER]

obtendremos el fatídico mensaje «**SINTAX ERROR**», porque en el vocabulario del ordenador no existe el comando PRINT 4.

La importancia de los espacios

De aquí se deduce que todas las palabras clave o comandos de Basic necesitan estar DELIMITADOS para que el AMSTRAD pueda reconocerlos. En palabras técnicas, el espacio que colocamos detrás del PRINT es precisamente eso: un DELIMITADOR, porque marca sus límites (*obviamente, si el comando se encuentra en el medio de una línea, necesitará un espacio por delante y por detrás para que el micro pueda «aclararse»*).

Escribir frases en pantalla era igualmente sencillo; así:

PRINT «MICROHOBBY AMSTRAD» conseguirá que la frase entre comillas se muestre en el monitor. Al contrario de los comandos, que pueden ir escritos en mayúsculas o minúsculas, lo que se escribe entrecomillado se mostrará literalmente así; por eso se habla de «strings literales»; por tanto, es innecesario decir lo que aparecerá en pantalla si tecleamos:

PRINT «MICROHOBBY AMSTRAD»

De paso, conviene comentar que en el caso de frases entrecomilladas el espacio después del PRINT es opcional. Las propias comillas cumplen la función de delimitadores.

Hasta ahora, nos hemos limitado a darle a la máquina una instrucción cada vez. Sin embargo, a poco compleja que sea la tarea que pretendemos realizar, será necesario dividirla en parte y darle al AMSTRAD muchas instrucciones. Por ejemplo, supongamos que queremos que aparezca en la pantalla el siguiente mensaje:

PROGRAMAR
ES
FACIL

Con nuestro método «**paso-a-paso**» tendríamos que teclear:

PRINT «PROGRAMAR» [ENTER]
PRINT «ES» [ENTER]
PRINT «FACIL» [ENTER]

Rápidamente se ve que de esta forma no funciona, porque cada instrucción sucesiva aparece en la pantalla «entre medias». Lo que necesitamos es darle al micro tres instrucciones tales como:

1. Escribe «programar»
2. Escribe «es»
3. Escribe «fácil»

en secuencia, de modo que el ordenador las ejecute sin detenerse a preguntar qué queremos que haga después.

Un programa es una secuencia de instrucciones

Tal secuencia de instrucciones es llamada PROGRAMA y, como podéis ver, va numerada en orden creciente, para que el micro pueda saber a donde debe dirigir su atención primero. Bien, pues vamos a escribir un programa cuya salida (*en inglés y en muchos libros*



Primeros pasos

de informática, «output» o «print out») sea:

PROGRAMAR
ES
FACIL

Para ello, vamos a asegurarnos antes de que estamos en modo 1 y de que la memoria del ordenador está limpia, libre de otros programas y datos. La forma de hacerlo es teclear:

MODE 1 [ENTER]
NEW [ENTER]

Ahora puede comenzar nuestro programa, tecleando:

10 PRINT «PROGRAMAR» [ENTER]

Hemos elegido para numerar la primera instrucción el número 10 en lugar del 1. En la práctica de la programación, se suele numerar las líneas de 10 en 10 por razones que más tarde os parecerán obvias. Llamamos al número de una instrucción su NUMERO DE LINEA.

EL ORDENADOR NO EJECUTA INMEDIATAMENTE LA INSTRUCCION. Ningún «PROGRAMAR» aparece en la pantalla. Esta reacción inesperada se debe al número de línea, el cual le informa a la máquina que lo que sigue no debe ser obedecido ahora, sino atesorado cuidadosamente en la memoria junto con el resto de las instrucciones que vengan después, si existen, para ser ejecutadas conjuntamente en el momento oportuno.

Acabamos, pues, de descubrir que todo ordenador posee dos MODOS DE OPERACION, uno, llamado MODO INMEDIATO, en el cual la orden es obedecida en cuanto se pulsa [ENTER] y un segundo, conocido como MODO PROGRAMA, en el que las instrucciones se depositan en memoria, esperando pacientemente a cobrar vida. Este es el que nos interesa.

Ahora, por favor, teclead:

20 PRINT «ES» [ENTER]
30 PRINT «FACIL» [ENTER]

y, en modo directo, CLS [ENTER], para borrar la pantalla.

Todo nuestro texto ha desaparecido, pero no hay que preocuparse demasiado. Existe un comando Basic que nos mostrará en pantalla nuestro programa completo, lo que se llama el LISTADO del mismo. Para verlo y creerlo:

LIST [ENTER]

y debería aparecer el programa número 1.

Los comandos aparecen en el listado siempre en mayúsculas

Si el comando PRINT se tecleó en minúsculas, ahora aparecerá en mayúsculas; en efecto, el AMSTRAD se encarga de que todas las palabras-clave, para distinguirlas mejor, sean

listadas en mayúsculas. Un simple y excelente método de saber si hemos cometido algún error de tipografía; si un comando no aparece en mayúsculas, algo ha fallado.

Ahora viene lo bueno. Todavía no le hemos ordenado al micro que ejecute el programa, se ha limitado exclusivamente a almacenar en su memoria una serie de instrucciones. Nuestro programa cobra vida mediante la orden RUN (en inglés, CORRER, y traducido a nuestra lengua como ejecutar).

Así:

RUN [ENTER]

«**echar a andar**» nuestro programita y veremos en pantalla el deseado:

PROGRAMAR
ES
FACIL

al menos, debiéramos verlo. Si no es así, sería conveniente inspeccionar el listado (LIST) para averiguar dónde está el fallo. Tal vez incluso se vea un mensaje de error concerniente a algún número de línea en particular.

De sabios es rectificar

En previsión de algún posible «despiste» en éste o posteriores programas, vamos a ver cómo podríamos corregirlo. Trataremos de alterarlo para escribir:

PROGRAMAR
ES
SIMPLE

es decir, sólo hay que modificar la línea número 30 del programa 1. Pues bien, el método más simple de todos (y tosco) es ¡teclarla de nuevo! La nueva versión reemplazará a la antigua en la memoria del AMSTRAD, y eso es todo; para demostrarlo, tecleese:

30 PRINT «SIMPLE» [ENTER]

a continuación LIST [ENTER] y, ¡hops!, tenemos el programa número 2. Como prueba final y decisiva, RUN [ENTER] mostrará en la pantalla:

PROGRAMAR
ES
SIMPLE

Un ejemplo más; supongamos que, por error, en la línea 10 del programa 2 escribimos:

10 PINT «PROGRAMAR» [ENTER]

Cuando tratemos de ejecutarlo (RUN) obtendremos el mensaje «**SINTAX ERROR IN**

10» (sólo al intentar ejecutar el programa, no al introducir la línea en memoria). Bien, simplemente volvemos a escribir la línea 10 correctamente y funcionará sin más problemas.

Hay formas más sofisticadas de corregir (**EDITAR**) una línea, pero las revisaremos posteriormente. De momento, éste será nuestro método «**antierrores**».

Naturalmente, si nos percatamos de un error de escritura MIENTRAS introducimos una línea de programa, usaríamos la tecla DEL para borrarlo, siguiendo a partir de ese punto hasta finalizar con ENTER la instrucción.

Hasta este momento, hemos ejecutado solamente dos programas; sin embargo, con ellos como modelo y un poco de imaginación, que sabemos no falta a ningún programador, se puede escribir virtualmente cualquier mensaje que se desee en la pantalla.

Sentando cátedra (con perdón)

Un punto importante acerca de este artículo y sucesivos es que proporcionaremos muchos programas, en general cortos, para telear y aprender de ellos. Todos poseen dos características comunes:

— Hacen incipiente en conceptos fundamentales en programación (*de lo contrario no estarían aquí*).

— Lo que aparece en pantalla, esto es, el output, suele ser bastante trivial y en muchos casos existirán maneras más simples de obtener el mismo resultado.

Programar es, en cierto sentido, como conducir; sólo se puede mejorar HACIENDOLO, no leyendo acerca de ello. Así que, por favor, ejecutad los programas ejemplos que se proponen por muy simples u obvios que parezcan. También, y esto SI que es importante, hay que procurar ir más allá de los programas ejemplo: modificarlos de todas las formas que se nos ocurran, dar campo libre a la imaginación. Sobre todo, es esencial mirar todo lo que aquí se dice con escepticismo, y no darlo por cierto hasta que personalmente se vea que efectivamente es así y el porqué de ello. Se aprenderá mejor y más rápido de los ejemplos que uno mismo haya desarrollado, en lugar de limitarse meramente a repetir punto por punto los nuestros.

Por tanto, parece una buena idea dedicar algo de tiempo a crear programas que escriban cosas en la pantalla de maneras distintas, porque para conseguirlo se necesitará emplear una sorprendente cantidad de comandos del Basic además de PRINT.

Así, sin darse cuenta.

Bueno, el sermón ha concluido; desde aquí nos parece escuchar los suspiros de alivio.

Supongamos que de nuevo queremos alterar el programa número 2 para que produzca como salida:

PROGRAMAR ES
BASTANTE SIMPLE

PROGRAMAS

```
5 REM *** PROGRAMA 1 ***
10 PRINT"PROGRAMAR"
20 PRINT"ES"
30 PRINT"FACIL"
```

```
5 REM *** PROGRAMA 2 ***
10 PRINT"PROGRAMAR"
20 PRINT"ES"
30 PRINT"SIMPLE"
```

```
5 REM *** PROGRAMA 3 ***
10 PRINT"PROGRAMAR"
20 PRINT"ES"
25 PRINT"BASTANTE"
30 PRINT"SIMPLE"
```

```
5 REM *** PROGRAMA 4 ***
10 CLS
20 PRINT"MICROHOBBY"
30 PRINT"AMSTRAD"
40 PRINT"SEMANAL"
```

```
5 REM *** PROGRAMA 5 ***
10 CLS
15 PRINT
20 PRINT"MICROHOBBY"
25 PRINT
30 PRINT"AMSTRAD"
35 PRINT
40 PRINT"SEMANAL"
45 PRINT
```

```
5 REM *** PROGRAMA 6 ***
10 CLS
20 PRINT"HOLA";
30 PRINT"QUE";
40 PRINT"HAY";
```

```
5 REM *** PROGRAMA 7 ***
10 CLS
20 PRINT"HOLA QUE HAY"
```

```
5 REM *** PROGRAMA 8 ***
10 PRINT"ME"
20 PRINT"DIVIERTE"
30 PRINT"PROGRAMAR"
40 GOTO 10
```

```
5 REM *** PROGRAMA 9 ***
10 CLS
20 PRINT"ESTO ES"
30 PRINT"MUY RARO"
40 GOTO 10
```

Obviamente, necesitamos una nueva línea de programa entre la 20 y la 30, así que podemos probar con:

25 PRINT «BASTANTE» [ENTER]

Si ejecutamos el comando LIST, nuestro programa se convierte en el número 3. Resulta que la línea 25 se ha ubicado correctamente a pesar de que la introdujimos después; en efecto, el AMSTRAD se encarga de ello. Ejecutando el programa se confirmará lo que digo.

La posibilidad de insertar nuevas líneas en un programa es la razón de escribir sus números de línea de 10 en 10, dejando un hueco para imprevistos cambios en el listado (*siempre son necesarios*).

Introducir el programa 4, recordando pulsar [ENTER] al terminar de escribir cada línea. Si después de listarla aparece una línea 25 fantasma, es que no se tecleó NEW antes de escribir el programa 4, con lo cual esa línea permanece agazapada lista para arruinar nuestra nueva creación. Moraleja: use new antes de introducir un programa en memoria.

Ahora bien, si cumplimos esta regla, el reñido nacido programa 4 «**morirá**»; ¿que hacer? No hay que preocuparse, simplemente tecleando:

25 [ENTER]

la malhadada línea desaparecerá del mapa. El AMSTRAD, cada vez recibe un número seguido de [ENTER], entiende que deseamos que olvide una línea de programa con ese número, si existe. Si no existe..., bueno, a lo mejor es interesante comprobar qué sucede en este caso.

El programa 5 muestra cómo escribir un mensaje separando las palabras mediante líneas en blanco (*línea 15, por ejemplo*).

Basta con una línea de programa que sólo contenga la sentencia PRINT.

Nuevos delimitadores para formatear la escritura

Con el programa 6, sin embargo, obtendrámos la salida:

HOLAQUEHAY

esto es, cada «**string**» aparece inmediatamente después que su antecesor. La labor del «;» es actuar como un «**pegamento**», impidiendo el paso a la línea siguiente. Como no hay espacios dentro de cada cadena (o string), las palabras forman un todo. Tratar de que el mensaje sea legible añadiendo los espacios apropiados a cada palabra. De paso, obsérvese que la misma salida puede obtenerse mediante el programa número 7. Un poco de práctica con el «;» hasta comprender plenamente su funcionamiento ayudará para posteriores ejemplos más complejos. Vamos a bregar ahora con algo completamente diferente

(programa número 8). El efecto es bastante impresionante, ¿verdad?

Hasta este momento, nuestros programas se limitaban a copiar en la pantalla el contenido entrecerrillado de las líneas; aquí, con la adición de una más, la cantidad de output (*salida en pantalla*) se incrementa considerablemente, y lo único que hemos hecho es ordenar al AMSTRAD que repita una y otra vez la misma secuencia de operaciones (*instrucción 40; «vete a la línea 10», en inglés GOTO 10*). Si la salida del programa va demasiado rápida, podemos detenerla pulsando la tecla ESC; el programa continuará ejecutándose a la pulsación de cualquier otra tecla excepto ESC, en cuyo caso se detendría, devolviéndonos el control.

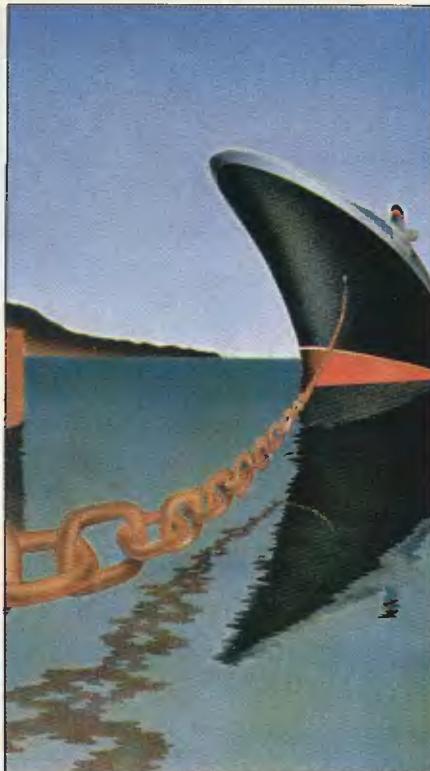
Los bucles: 1.ª clave de la programación

Lo que sucede en el interior del ordenador es lo siguiente:

ME (línea 10)
DIVIERTE (línea 20)
PROGRAMAR (línea 30)

y entonces llega a la línea 40 que le ordena volver a repetir el ciclo, observad cómo al «**acabarse**» la pantalla, el AMSTRAD, él solo, hace sitio para más texto por el expeditivo procedimiento de subir hacia arriba el existente.

El nombre que se emplea para indicar la condición de que una serie de líneas de programa se repitan un cierto número de veces



es BUCLE; así, hablamos de que el programa número 8 tiene un bucle entre las líneas 10 y 40. Decimos, además, que el bucle es INCONDICIONAL, esto es, no le hemos dado al ordenador instrucción alguna que permita detener el programa cuando determinado hecho suceda (*cómo pulsar una tecla en concreto, por ejemplo*). Esta no es una buena práctica de programación que digamos, ¡los micros no suelen tener la cortesía de abandonar ellos mismos un bucle sin fin!

Para detener **ALGUNOS** de los tipos de bucles incondicionales, tenemos que interrumpirlos «desde fuera», bien mediante la tecla ESC, bien pulsando CTRL (y *mantiéndolo pulsado*) junto con SHIFT y ESC. Este segundo método es bastante radical y no muy aconsejable, pues causa lo que se conoce como RESET, la vuelta del ordenador a su estado recién encendido, con la consiguiente pérdida de cualquier programa que guardará en su memoria. Por el contrario, ESC detiene el programa y nos da la opción de continuar pulsando una tecla que no sea la misma ESC, en cuyo caso el programa en curso se paraliza definitivamente, pero sin desaparecer de la memoria.

Los que quieran divertirse un poco experimentando con bucles sin fin, Prueben a crear un programa que escriba repetidamente en pantalla la siguiente fecha formada por asteriscos:

```
*
 ***
 ****
 *****
```

Por último, o casi, **¿alguien podría decirnos qué anda mal en el programa número 9?** Gracias.

La próxima semana comenzaremos a usar variables para incrementar la potencia de nuestros programas.

MICROHOBBY AMSTRAD SEMANAL

LE OFRECE AHORA SUS PROGRAMAS YA GRABADOS, PARA QUE VD. NO TENGA QUE TECLEARLOS

Todos los programadores y aficionados a la microinformática sabemos lo tedioso y propenso a errores que resulta teclear un listado de un programa. Para facilitar tu labor al máximo y que no tengas que estar horas sobre el teclado de tu ordenador tratando de descifrar incomprensibles mensajes de error, **AMSTRAD SEMANAL** te ofrece cada mes los programas publicados de los cuatro números correspondientes en una cinta de cassette, sólo por **675 ptas. (sin más gastos por envío)**.



Para que tus dedos no realicen el trabajo duro, M.H. AMSTRAD lo hace por ti. Todos los listados que incluyan este logotipo se encuentran a tu disposición en un cassette mensual, solicitánselo.

Envianos con la menor demora posible, el cupón correspondiente.

BIORRITMOS

Este programa ayudará a todas las personas interesadas a ver, instantáneamente, sus biorritmos de cualquier mes (y año), de su vida pasada, presente y futura.

Los biorritmos son curvas que reflejan, o así lo aseguran sus defensores, los ciclos vitales que atraviesa nuestro cuerpo, influyendo poderosamente en nuestro comportamiento diario a nivel físico, intelectual y emocional.

El ciclo intelectual se repite cada 33 días, mientras que los períodos de los otros dos son más cortos: 28 para el emocional y sólo 23 para el físico.

Cuando el programa arranca, se nos pregunta por nuestro nombre, día de nacimiento y el mes cuyos biorritmos queremos ver en la pantalla.

Por cierto, lo que se ve en ella es un calendario dibujado en perspectiva, con una torre de tres colores para cada día del mes. La correspondencia entre colores y ciclos es la siguiente:

- Azul, intelectual.
- Verde, físico.
- Marrón, emocional.

Un trazo delgado de color representa que la cosa va mal. Un trazo más grueso, bien.

Uno puede planear sus actividades diarias de acuerdo con sus biorritmos, si así lo desea.

Si vamos a correr el marathon, bien, sería conveniente elegir un día en el que nuestro nivel de verde sea muy, muy alto.

Los aficionados al ajedrez o al Mastermind o a algo similar, ganarán sin duda cuando la torre correspondiente al día del juego «chille» de azul.

En cuanto al sexo, no está muy claro si ésta interesante actividad cae bajo el ámbito de lo físico o de lo emocional. Cúrese en salud, ¡y comprobaré ambos!

Ocasionalmente verá que hay días en los que las torres son apenas un débil trazo de marrón: fatal. Todo lo que se haga en esos días acabará en un completo desastre. Quédese en la cama y no haga nada.

El programa está provisto de sentencias REM en aras de la claridad, y se ha procurado estructurarlo para que su comprensión sea fácil.

El bucle principal comienza en la línea 100 y llama a la rutina de entrada de datos para averiguar nuestro nombre, fecha de nacimiento y el mes que nos interesa. Entonces viene otra llamada a la rutina de «calendario».

Al finalizar de estudiar el biorritmo del mes elegido, podemos ver el siguiente (pulsando «P»), el anterior («A») o volver al menú («M»).

VARIABLES PRINCIPALES

NOMBRE FUNCION

bd,bm,by,bz	Fecha de nacimiento: día, mes, año y número de días desde el 31 de diciembre.
dm,dy	Mes en pantalla.
names\$	Nuestro nombre.
mth\$	Matriz que almacena los nombres de los meses.
wkdy\$	Nombres de los días de la semana.
do	Primer día del mes.
dn	Número de días por mes.
wn	Número de semanas por mes.
x0,y0	Esquina inferior izquierda de la base del calendario.
xg,yg	Esquina superior izquierda.
xt,yt	Posición de cada torre.
t	Altura de la torre.
d,m,y,z	Variables de trabajo para datos.
a,a\$,b,b\$	Variables de uso general.



Serie Oro



Las fechas se introducen en el formato 28nov44, por ejemplo, con o sin espacios. Se decodifican en la rutina localizada entre las líneas 910-1030. La variable «m» se pone a cero si las 3 letras del nombre del mes no se reconocen. A los años menores que 100 se les suman 1900.

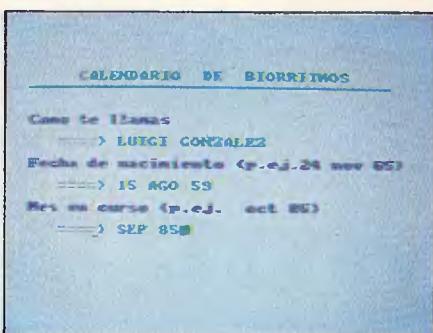
La función INSTR de la línea 130 coge la tecla pulsada y la compara con ESPACIO, P, A o M, produciendo un número que será utilizado por la sentencia ON GOTO.

PRINT CHR\$(23) y PRINT CHR\$(0) en las líneas 400 y 1310 colocan el modo gráfico opaco, mientras que CHR\$(23) y CHR\$(3) en la línea 1230 lo cambia a modo «OR» para evitar sobrescribir la pantalla.

Sin embargo, CHR\$(23) en 1190 es un truco para suprimir un espacio no deseado.

El texto que se escribe en líneas 1230 y 1310 se coloca en la posición del cursor gráfico.

Tanto si se es creyente en el tema de los biorritmos como si no, no cabe duda que disfrutaréis con la visión tridimensional que proporciona el programa, con su presentación soberbiamente lograda. Además, la detallada inspección del listado, permitirá al programador ávido aprender más de una técnica eficaz y sutil.



```

10 REM *****
20 REM * CALENDARIO BIORRITMICO *
30 REM * -----
40 REM * Tony Forbes *
50 REM *****
60 REM
70 REM
80 REM
90 GOSUB 310:REM inicializacion
100 REM *** BUCLE PRINCIPAL ***
110 GOSUB 660:REM carga de datos
120 GOSUB 190:REM calendario
130 ON INSTR(" PAM",UPPER$(INKEY$))+1 GD
TO 130,130,140,160,110
140 dn=dm MOD 12+1:dy=dy-(dn=1)
150 GOTO 120
160 dn=(dn+10) MOD 12+1:dy=dy+(dn=12)
170 GOTO 120
180 REM *** calendario ***
190 MODE 1
200 do=FNwkday(FNz(1,dm,dy))
210 dn=FNdmth(dm,dy)
220 wn=INT((dn+do+6)/7)
230 xo=32*wn:yo=56*wn
240 GOSUB 1050:REM parrilla
250 xt=xo-22+xg:yt=yo-38+yg
260 GOSUB 1150:REM text
270 xt=xt+22:yt=yt+4
280 GOSUB 1370:REM torres
290 RETURN
300 REM *** inicializacion ***
310 DEFINT a-y
320 GOSUB 470:REM funciones
330 BORDER 16
340 INK 0,16:INK 1,9:INK 2,1:INK 3,3
350 RESTORE 380
360 DIM mth$(12)
370 FOR a=1 TO 12:READ mth$(a):NEXT
380 DATA Ene,Feb,Mar,Abr,May,Jun,Jul,Ago
,Sep,Oct,Nov,Dic
390 wkds="Dom Lun Mar Mie Jue Vie Sab"
400 PRINT CHR$(23);CHR$(0);
410 xo=8:yo=16
420 name$="Tamsin"
430 bd=3:bm=5:by=1980

```

```

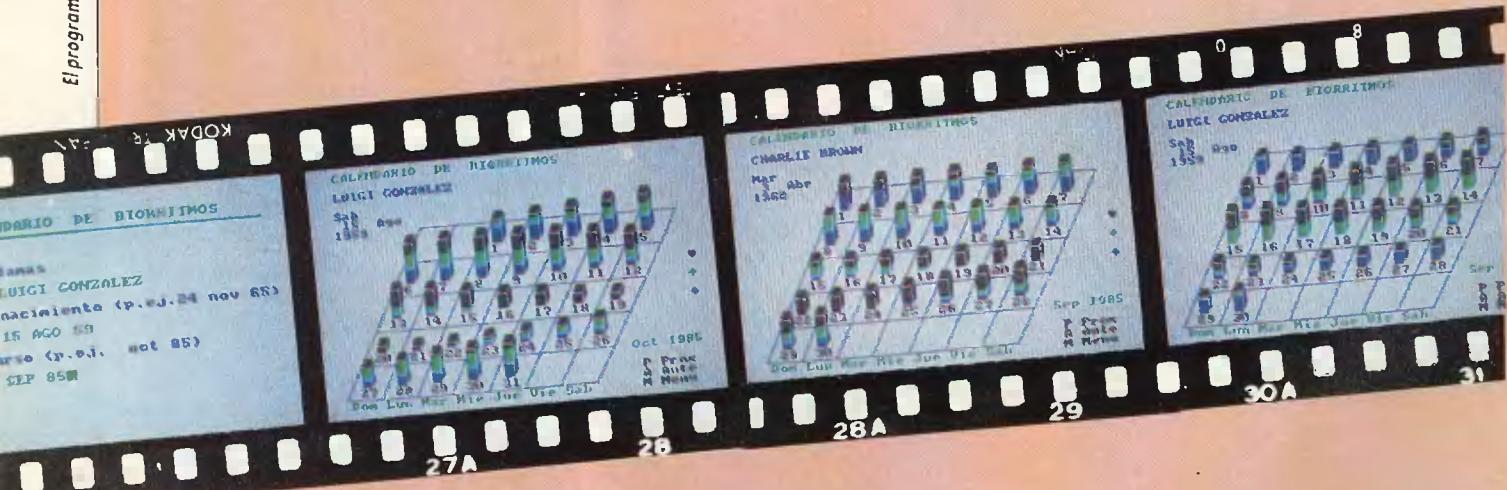
440 dn=4:dy=1985
450 RETURN
460 REM *** FUNCIONES ***
470 REM y --> CIERTO si y es bisiesto
480 DEF FNleap(y)=y MOD 4=0 AND (y MOD 1
00)<0 OR y MOD 400=0
490 REM m,y --> numero de dias del mes
500 DEF FNdmth(m,y)=INT(31+COS(2.7*(m-7.
5))+*(m-2)*(2+FNleap(y)))
510 REM d,m,a --> numero de dias despues
de Cristo
520 DEF FNz(d,m,y)=y*365+*(y+3)\4-(y+99)\1
00+*(y+399)\400+INT(38.401*(m-1))-(m-2)+
(m-2) AND m<(8)-(m-2) AND FNleap(y)+d
530 REM d,m,y --> CIERTO si la fecha es
valida
540 DEF FNok(d,m,y)=(y)=0 AND y<=9999 AN
D m>0 AND d>0 AND d<=FNdmth(m,y)
550 REM z --> dias de la semana 0=Dom, 1
=Lun, etc
560 DEF FNwkday(z)=z+5-INT((z+5)/7)*7
570 REM calendario de nivel
580 DEF FNlev(z,p)=INT((SIN((z/p-INT(z/p
)))*2*PI)+1)*6.999
590 REM posicion en el calendario
600 DEF FNxp(a)=xt+64*(a MOD 7)-32*(a\7)
610 DEF FNyp(a)=yt-56*(a\7)
620 RETURN
630 REM *** carga de datos ***
640 MODE 1:WINDOW 3,38,4,25
650 LOCATE 6,1:PRINT "CALENDARIO
DE BIORRITMOS":PRINT STRING$(35,154)
660 LOCATE 1,5:PRINT "Como te llam
as";
670 LOCATE 4,7:GOSUB 880:REM input
680 IF a$="" THEN 780
690 name$=LEFT$(a$,20)
700 LOCATE 1,9:PRINT "Fecha de nac
imiento (p.ej.24 nov 85)";
710 LOCATE 4,11:GOSUB 880:REM input
720 IF a$="" THEN 780
730 GOSUB 920:REM comprueba datos
740 IF NOT FNok(d,m,y) THEN PRINT " Dato
750 GOSUB 920:REM comprueba datos
760 IF NOT FNok(d,m,y) THEN PRINT " Dato

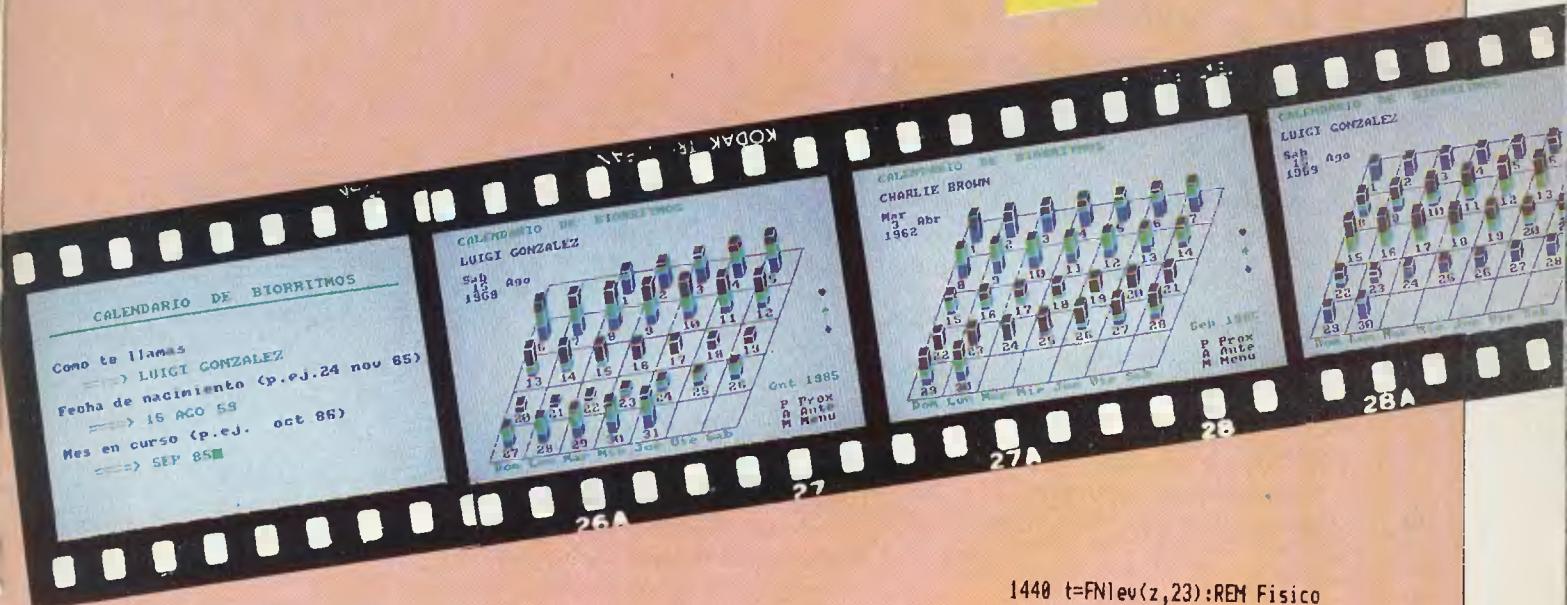
```

```

s incorrectos prueba otra vez";:GOTO 730
770 bd=d:bm=n:by=y
780 zb=FNz(bd,bm,by)
790 LOCATE 1,13:PRINT "Mes en curs
o (p.ej. oct 85)";
800 LOCATE 4,15:GOSUB 880:REM input
810 IF a$="" THEN RETURN
820 a$="1"+a$
830 GOSUB 920:REM comprueba datos
840 IF NOT FNok(d,m,y) THEN PRINT " Dato
s incorrectos prueba otra vez";:GOTO 800
850 dn=n:dy=y
860 RETURN
870 REM *** input
880 PEN 1:INPUT "==== ",a$
890 LOCATE 1,18:PRINT CHR$(20);
900 RETURN
910 REM *** convierte datos a$ --> d m a
920 d=VAL(a$)
930 a$=UPPER$(a$)
940 FOR a=2 TO LEN(a$)
950 IF INSTR("EFMAJSOND",MID$(a$,a,1))>
THEN 970
960 NEXT
970 b$=MID$(a$,a,3)
980 FOR m=12 TO 0 STEP-1
990 IF UPPER$(mth$(m))=b$ THEN 1010
1000 NEXT
1010 y=VAL(MID$(a$,a+3))
1020 y=y-1900*(y<99)
1030 RETURN
1040 REM *** parrilla ***
1050 MOVE xo,yo
1060 FOR a=0 TO wn
1070 DRAWR 448,0,2:MOVER -448+32,56
1080 NEXT
1090 FOR a=0 TO 7
1100 MOVE xo+64*a,yo
1110 DRAWR xo,yg
1120 NEXT
1130 RETURN

```





```

1140 REM *** texto ***
1150 PEN 1:LOCATE 1,1:PRINT "CALENDARIO
DE BIORRITMOS"
1160 PEN 2:PRINT:PRINT name$
1170 PRINT:PRINT MID$(wkd$,4*FNwkday(zb)
+1,3)
1180 PRINT bd;" ";mth$(bm)
1190 PRINT CHR$(23);by
1200 WINDOW #1,33,40,21,25
1210 PEN #1,1:PRINT #1,mth$(dm);STR$(dy)
1220 PEN #1,3:PRINT #1," P Prox A Ante
M Menu ";
1230 PRINT CHR$(23);CHR$(3);:TAG
1240 PLOT -8,0,1
1250 MOVE xo+8,yo-4

```

```

1260 PRINT wkd$;
1270 PLOT -8,0,3
1280 FOR d=1 TO dn
1290 MOVE FNxp(d+do-1),FNyp(d+do-1)
1300 PRINT d;:NEXT
1310 TAGOFF:PRINT CHR$(23);CHR$(0);
1320 LOCATE 40,12:PEN 3:PRINT CHR$(228);
1330 LOCATE 40,14:PEN 1:PRINT CHR$(229);
1340 LOCATE 40,16:PEN 2:PRINT CHR$(227);
1350 RETURN
1360 REM *** torres ***
1370 z=FNz(0,dm,dy)-zb
1380 FOR d=do TO do+dn-1:z=z+1
1390 MOVE FNxp(d),FNyp(d)
1400 PLOT 0,0,2
1410 t=FNlev(z,33):REM Intelectual
1420 GOSUB 1590:REM nivel
1430 PLOT 0,0,1

```

```

1440 t=FNlev(z,23):REM Fisico
1450 GOSUB 1590:REM nivel
1460 PLOT 0,0,3
1470 t=FNlev(z,28):REM Emocional
1480 GOSUB 1590:REM nivel
1490 FOR a=0 TO 3
1500 DRAWR 16,0:MOVER 0,2
1510 DRAWR -16,0:MOVER 2,2
1520 NEXT
1530 MOVER -8,-16
1540 DRAWR 16,0,0
1550 DRAWR 6,14:MOVER -6,-14
1560 DRAW FNxp(d)+16,FNyp(d)
1570 NEXT:RETURN
1580 REM *** nivel ***
1590 FOR a=1 TO t
1600 DRAWR 16,0
1610 DRAWR 6,14
1620 MOVER -22,-12
1630 NEXT
1640 RETURN

```

FUNCIONES PRINCIPALES

NOMBRE	FUNCION
FN1 eap(y)	Retorna —1 si el año es bisiesto, 0 si no lo es.
FNdmt(m,y)	Reconoce correctamente el fin de cada siglo.
FNZ(d,m,y)	Calcula el número de días de un mes.
	Devuelve el número de días desde el 31 de diciembre. Por simplicidad, se ha ignorado el cambio producido por el calendario Gregoriano, por lo que fechas anteriores al siglo 16 pueden dar resultados erróneos.
FNoK(d,m,y)	Retorna —1 si la fecha es válida.
FNwkday(z)	Retorna el día de la semana número «z» (resultante de la función FNz). 0 si es domingo, 1 si es lunes, etc.
FNth\$(a)	Devuelve el par de letras correcto para unir al final del número «a».
FN1 ev(z,p)	Nivel del período «p» para el día «z».
FNxp,FNyp	Donde construir una torre.



P
ara que tus dedos
no realicen el trabajo duro, M.H. AMS-
TRAD lo hace por ti. Todos los listados que incluyen
este logotipo se encuentran a tu disposición en un cas-
sette mensual, solicítanoslo.

DECIMALES ENCOLUMNADOS

Análisis

Colocar una serie de números decimales, de forma que los puntos decimales queden en la misma columna, es un problema de fácil solución con ayuda de los comandos LEN y MID\$, los cuales son utilizados para detectar el punto decimal.

NUMERO DE LINEA

- 10, 20 Estas sentencias REM contienen el nombre del programa, con los asteriscos que separan éste del resto del listado. El AMSTRAD ignora todo lo que se encuentra después de un REM.
- 30 Pone el ordenador en MODO 2, de 80 caracteres por columna.
- 40-100 Constituyen un bucle FOR NEXT, que efectúa diez iteraciones para tratar los cinco números contenidos en el DATA de la línea 200.
- 50 Cada vez que se repite una iteración, el bucle lee el siguiente número de los DATA y lo almacena en la variable alfanumérica NUMERO\$.
- 60 Pone a cero el valor de POSICION, en cada iteración del bucle.
- 70 Efectúa una llamada a la rutina

80

na que fija la posición del punto decimal contenido en NUMERO\$.

90

Se encarga de colocar todos los puntos decimales en la misma columna, lo cual se logra restándole a 40 el número de cifras que se encuentran a la izquierda del punto decimal.

110

Representa un número en pantalla, cada vez que se efectúa una iteración.

120-180

Detiene la ejecución del programa, antes de que éste invada la zona de subrutinas sin ser llamadas.

130-170

Forman la subrutina, que se encarga de buscar los puntos decimales en la variable NUMERO\$.

140

Bucle FOR NEXT, con tantas iteraciones como caracteres tiene la variable NUMERO\$.

Toma un carácter de la varia-

150

ble NUMERO\$, y lo almacena en CHECK\$. Esta operación se repite, hasta que todos los caracteres contenidos en NUMERO\$, son almacenados en CHECK\$.

160

Comprueba si CHECK\$ contiene algún punto decimal, cuando encuentra alguno, guarda en POSICION el valor contenido en BUSCA, el cual marca la posición del punto decimal.

180

Si una vez comprobada toda la cadena de la variable CHECK\$, no encuentra punto decimal, suma uno a la variable POSICION, de forma que el número queda todo a la izquierda de un punto imaginario.

200

Final de la subrutina, devuelve el control a la línea 80.

Contiene los diez números que queremos encolumnar.



567.00
34.354
1105
745.23
0.723
45678.23
12.90
7.32
976.567489
2300.4563223

números
encolumnados tras la
ejecución del
programa

Suscríbete... y uno de estos tres sensacionales juegos será tuyo... ¡GRATIS!

M.H. AMSTRAD te da a elegir entre tres de los mejores juegos existentes en el mercado para **AMSTRAD**; **POLE POSITION**, **DALEY THOMPSON'S DECATHLON** y **BEACH HEAD**, cualquiera de los cuales puede ser tuyo solamente con suscribirte a nuestra revista. Aprovecha esta ocasión excepcional y ahorra 2.100 pesetas (precio de venta del programa) más el importante descuento que se produce en el precio de cada número, por el hecho de ser suscriptor. Disfruta de las ventajas que supone recibir cómodamente tu revista a domicilio y de la seguridad de tener tu ejemplar aunque se haya agotado en los quioscos.

Enviános tu boletín de suscripción y no le des más vueltas, el número de juegos para regalos de suscripción aunque grande, es limitado y éstos se podrían agotar mientras lo estás pensando.

BEACH HEAD producido por U.S. GOLD es una misión de desembarco en una costa fuertemente defendida por las fuerzas aeronavales enemigas. Debes conducir tu flota hacia la bahía y repeler el ataque aéreo, si lo consigues tu siguiente obstáculo será una flotilla de destructores y acorazados, superada la cual desembarcarás tus anfibios en las arenas de la bahía, éstos deben superar las defensas costeras y llegar a la fortaleza que es el objetivo final.

POLE POSITION es la última creación de DATSOFT en la que podrás experimentar toda la emoción de un gran premio automovilístico, vuelta de clasificación y carrera cronometrada contra tus adversarios.

DALEY THOMPSON'S DECATHLON con este juego OCEAN enciende la llama olímpica y te reta a superar los récords de los campeones de todos los tiempos, el decathlon se desarrolla en dos días de competición y se compone de las siguientes pruebas:

PRIMER DIA: 100 m lisos, salto de longitud, lanzamiento de peso, salto de altura y 400 m lisos.

SEGUNDO DIA: 110 m vallas, lanzamiento de disco, salto con pértiga, lanzamiento de jabalina y los 1.500 m.

Utiliza el cupón adjunto a la revista o suscríbete por teléfono
(91) 733 50 12
(91) 733 50 16



Los REGISTROS (I)

La semana pasada pudimos examinar con cierto detalle los números hexadecimales y ejecutar dos programas de código máquina muy simples. Recordemos que para contar en hexa íbamos desde 1 hasta F, y prefijábamos nuestros números con el signo «&» para evitar confundirlos con los decimales.



Ara pasar de &F (15), último «dígito» de las unidades a las decenas, simplemente sumábamos 1 de la siguiente manera:
&F+1=&10

También descubrimos que son suficientes 4 dígitos hexadecimales para cubrir todo el espacio de direcciones del Z80. Así, 65536, máximo valor que el AMSTRAD puede manipular, se escribe en hexa &FFFF.

Estas cuatro cifras las dividimos en dos configuraciones más simples: lo que se conoce como byte alto (cuyos valores de columna son 4096 y 256 en decimal) y, cómo no, el llamado byte bajo (valores de columna 16 y 1 en decimal). Por ejemplo, &CDEF tiene &CD como byte alto y &EF como bajo.

La rutina que estudiamos hace poco era:

```
CD DB BB      CALL &BBDB
C9             RET
```

donde las palabras de la segunda columna son los mnemónicos que se usan como recordatorios de lo que su equivalente en números (columna izquierda) quieren decir.

Los mnemónicos son conocidos también como lenguaje ensamblador, y es prácticamente la única manera realista de programar en código máquina, aunque nosotros os propondremos hoy otra forma alternativa de hacerlo, más útil quizás para el principiante.

De momento, sabemos ya que la llamada a la DIRECCIÓN DE MEMORIA &BBDB borra la ventana de gráficos en uso, que, asumiendo que el ordenador acaba de encenderse, será toda la pantalla y no producirá un efecto muy espectacular.

Hay, dentro del firmware, otra rutina equivalente, para la pantalla de texto, que comienza en la dirección &BB6C.

Análogamente, se borrará toda la pantalla suponiendo que esté en uso la ventana por defecto del sistema.

El código necesario para que la rutina trabaje, sería:

CD 6C BB C9

o bien, en ensamblador, con los correspondientes códigos de operación:

```
CD 6C BB      CALL &BB6C
C9             RET
```

Para introducirla en la memoria del ordenador:

```
POKE &3000, &CD
POKE &3001, &6C
POKE &3002, &BB
POKE &3003, &C9
```

y para ejecutarla, CALL &3000.

La pantalla se borra, y, para demostrar que es la de texto, el mensaje «READY» aparece en la parte superior.

Cargador hexadecimal: un primer paso

Esta forma de ejecutar código máquina no resulta un buen sistema en cuanto nuestros programas sean un poquito largos. En efecto, sería terriblemente tedioso teclear el código mediante POKE byte a byte.

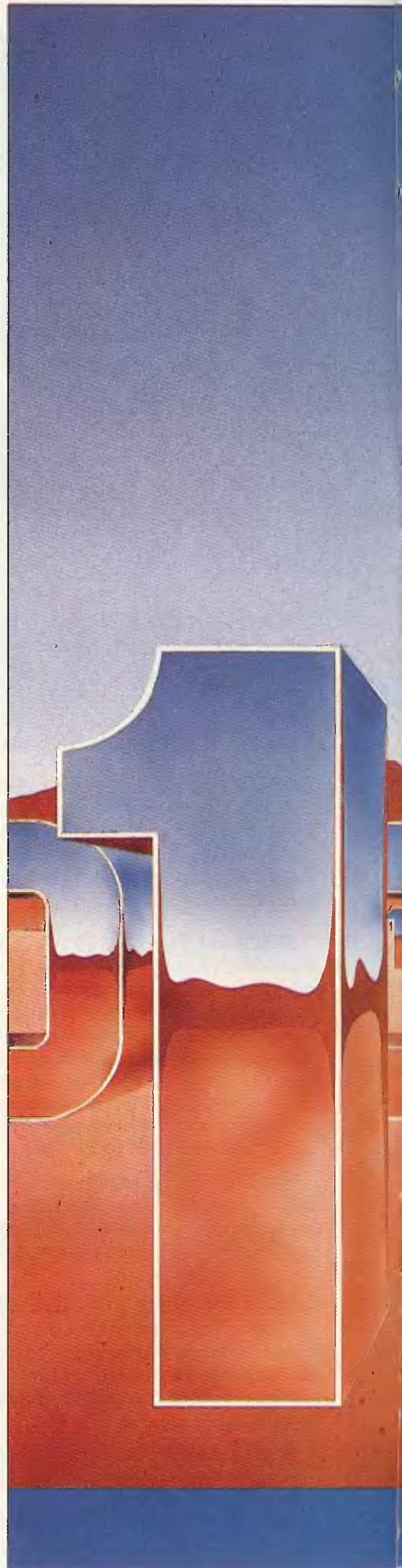
La solución viene dada en la forma de lo que se conoce «oficialmente» como cargador hexadecimal, el programa número 1.

Su función es atesorar en memoria los números exadecimales que nosotros escribimos.

Es esencial que el «HEXER» se conserve en cinta/disco para poder seguir satisfactoriamente estos artículos. Por tanto, sería conveniente teclearlo cuidadosamente y, sobre todo, GRABARLO ANTES DE EJECUTARLO.

Cuando tecleemos RUN, veremos en pantalla el siguiente menú de opciones:

1. Introducir código
2. Examinar código
3. Alterar código



Código máquina

4. Ejecutar código

5. Salir del programa

Para elegir entre ellas, simplemente pulsamos un número del 1 al 5.

Supongamos que queremos introducir, en hexadecimal, un programa; pulsaríamos el 1 y aparecerá en la pantalla el mensaje:

¿DIRECCIÓN DE COMIENZO?

Obviamente, el programa necesita saber dónde colocar el código máquina que vamos a introducirle, el equivalente de la dirección &3000 de la rutina anterior. Hexer, por razones que explicaremos más tarde, está preparado para bregar con código que comience en dicha dirección.

Por supuesto que podemos escoger la dirección de comienzo que nos acomode, pero de momento es mejor pulsar simplemente ENTER con lo que se tomará la dirección &3000 por defecto.

Acto seguido, leeremos:

¿byte?

Aquí el programa espera pacientemente un número hexadecimal, que será almacenado en memoria a partir de la dirección de comienzo elegida. Para borrar la pantalla de texto debemos introducir (también leeréis en algún que otro libro «entrar») los siguientes bytes:

CD 6C BB C9

Por tanto, a la pregunta de «¿byte?» responder con:

CD + ENTER

Obsérvese la ausencia del signo «&»; no es necesario. Hexer ya sabe que es un número hexadecimal.

En este punto, tenemos almacenado en &3000 &CD, es decir, en una terminología a la que debemos irnos acostumbrando. EL CONTENIDO DE LA DIRECCIÓN &3000 ES &CD.

De nuevo repetiríamos el ciclo con todos los bytes de nuestra rutina hasta llegar a (*e introducir*) &C9 + ENTER.

El programa en código máquina se encuentra ahora en la memoria. Pulsando «S» volvemos al menú principal).

Es esencial comprobar la corrección del código antes de ejecutarlo

Resistamos la tentación de ejecutar el programa y sigamos al sana costumbre de comprobar que el código máquina introducido es correcto; si por desgracia nos hemos equivocado,

cado, no tendremos una segunda oportunidad. El AMSTRAD se bloqueará y habremos perdido todo nuestro esfuerzo.

Para esto está la opción 2 del programa; elijámosla y, cuando nos pregunte la dirección de comienzo, pulsar ENTER una vez más. Veremos en la pantalla algo parecido a esto:

2FF8 31 0 34 8 0 0 0 0

Es decir, el programa muestra una secuencia de bytes de 8 en 8, indicando primero la dirección de memoria desde la que se parte (2FF8).

Así, en este caso, sabemos que &2FF8 contiene &31 (los números están en hexa), &2FF9 un 0, &2FFA (*¡jojo!, después del 9 viene la A*) &34, etc., hasta &2FFF que contiene un 0.

La siguiente dirección que veremos al pulsar la barra espaciadora es &3000 (&2FFF + 1 = &3000) y siguientes:

3000 CD 6C BB C9 0 0 0 0

donde los primeros 4 bytes, como cabría esperar, contienen nuestra rutina de borrado.

Comprobad cuidadosamente que estos bytes coinciden con los que habéis tecleado; si no es así, pulsar «S» para volver al menú y, con la opción 1 reintroducir el código máquina. Sólo son 4 bytes, así que no será un gran esfuerzo.

La razón de escoger &2FF8 como dirección por defecto para mostrar el contenido de la memoria, en lugar de la &3000, que es donde comienzan los programas, es que vamos a emplear estas posiciones como block de notas para mostrar los resultados de los mismos.

Bien, supongamos que el código está correctamente colocado y comprobado, y hemos vuelto al menú (*pulsar «S»*) principal.

La opción 4 nos permitirá ejecutar el programa, tras pulsar ENTER para indicar al programa que vamos a usar la dirección &3000 por defecto, y la pantalla debiera borrar rápidamente. Ejecutar un programa en código máquina es el último recurso para averiguar si funciona. Antes deben ser hecho todos los chequeos imaginables y algunos más; toda precaución es poca.

Hexer permite alterar código con toda facilidad

Por esta razón, si de pronto se cambia de idea, pulsando «X» volveremos al menú principal sin que un solo byte sea ejecutado. Otra posibilidad que un cargador hexa debe poseer es la de alterar el contenido de una posición de memoria, la opción 3 en el Hexer. Aparecerá en pantalla:

¿Alterar desde?

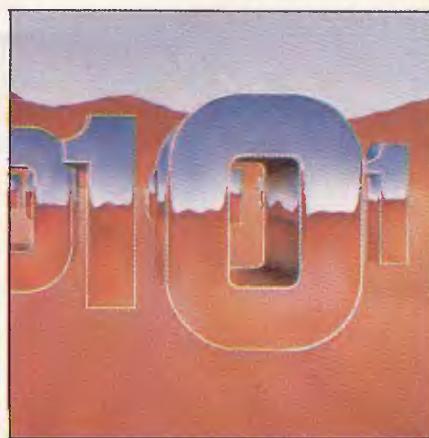
como siempre, ENTER presupone &3000, y cualquier otro número se aceptará como dirección de comienzo.

Supongamos que queremos transformar la rutina de borrado y, en lugar de:

CD 6C BB C9

queremos que sea:

CD DB BB C9



En la pantalla nos está aguardando desde que pulsamos ENTER:

3000 CD?

Como no queremos alterar este byte, ENTER de nuevo. El siguiente es el que nos interesa, por lo que cuando aparezca:

3001 6C?

tecleamos DB+ENTER. A partir de ese momento, el contenido de &3001 es &DB.

Cuando veamos la siguiente dirección, como ya no interesa alterar más código, «S» volverá al menú.

Resumiendo:

1. Para dejar un byte inalterado, pulsar ENTER.
2. Para alterarlo, escribir el nuevo valor en hexa (*sin el signo «&»*).
3. Pulsar «S» para volver al menú.

En el primer capítulo de estas series hablábamos de que el código máquina no era más que una serie de números almacenados secuencialmente en la memoria, y que todo el trabajo que había que hacer para escribir un programa con ellos era desplazarlos adecuadamente de un lugar a otro. También quedó clara la necesidad de **«controladores de tráfico»** para evitar atascos en la memoria, los llamados REGISTROS. Así, movíamos datos de la memoria a los registros y de ellos a otras posiciones de la memoria.

Los registros controlan el tráfico de bytes entre las memorias

Uno de los registros con más solera del Z80 es el A o acumulador; se encuentra en el corazón del microprocesador y sólo puede almacenar un byte. La utilidad del acumulador estriba en que podemos hacer cosas con un byte en su interior que no están permitidas cuando dicho byte se encuentra en memoria **«normal»**.

Teniendo en cuenta que el acumulador será un registro de uso ineludible en cualquier aplicación en código máquina, vamos a realizar un par de trucos con él; nada sofisticado, pero sí muy útil.

El primero de ellos envuelve una importante rutina del firmware, llamada por AMSTRAD

«TXT-OUTPUT» y cuya dirección de llamada es &BB5A.

La función que cumple es escribir en pantalla el carácter cuyo código ASCII se encuentra... ¡en el acumulador!

Introducir un número en el acumulador no es una tarea muy difícil. Su mnemónico es:

LD A,N

en donde **«LD»** es la abreviatura inglesa de **«cargar»** (*Load*), y **«n»** es, por supuesto, el número con el que cargamos el acumulador.

Por tanto:

LD A,&2A

introducirá el número &2A (decimal 42) en el registro acumulador. El opcode (*código de operación*) de **«LD A,N»** es &3E, y el Z80 espera encontrar inmediatamente después el valor **«N»**.

Expresado en bytes, la secuencia que tenemos que teclear en el Hexer es:

3E 2A

y, al ser &2A el código ASCII para el **«*»**, la llamada a continuación a &BB5A pondrá un asterisco en pantalla.

Necesitamos ordenar al AMSTRAD que haga lo siguiente:

- a) Cargar (*Load*) el acumulador con &2A.
- b) Llamar a TXT-OUTPUT (*dirección &BB5A*) para que el **«*»** aparezca en pantalla.

- c) RETornar desde esta rutina para recuperar el control del ordenador.

Asumiendo que el código máquina se cargará en la dirección &3000, estas órdenes se traducen en:

DIRECCIÓN	OPCODE	MNEMONICO
&3000	3E 2A	LD A, &2A
&3002	CD 5A BB	CALL &BB5A
&3005	C9	RET

Para crear esta rutina vía Hexer, tenemos que introducir, mediante la opción 1 en la dirección por defecto &3000, la serie de bytes:

3E 2A CD 5A BB C9

Como se puede ver, es simplemente el listado hexa de los códigos de operación. Revisese cuidadosamente y, si todo es correcto pues a ejecutarlo con la opción 2 del Hexer.

Si vemos en la pantalla un asterisco **«inesperado»** acabamos de crear, y hacer funcionar, el primer programa en código máquina en el que NOSOTROS decidimos qué va a pasar.

Modificar un programa mediante los registros

Obviamente, si el acumulador se carga con otro número, veremos otro **«invitado»** diferente en el display. Por ejemplo, para que aparezca una **«A»**, en lugar del **«*»**, nuestro programa tendría este aspecto:

DIRECCION	OPCODE	MNEMONICO
&3000	3E 41	LD A, &41
&3002	CD 5A BB	CALL &BB5A
&3005	C9	RET

En lugar de empezar desde el principio con la nueva rutina, pasamos a la opción 3 del Hexer, «Alterar código», y escogemos como dirección de comienzo &3001, ya que aquí está el byte que hemos de cambiar. Así, al «prompt»:

3001 2A ?

respondemos tecleando 41 (**y ENTER, por supuesto**) y luego pulsando «S» para volver al menú.

Si examinamos memoria, la serie correcta de bytes es:

3000 3E 41 CD 5A BB C9

Al ejecutar el programa, la «A» (código ASCII 65 decimal, &41 hexa) se mostrará en pantalla. Tratad, por favor, de conseguir que el programa pinte otras letras, mirando la tabla de códigos ASCII en el manual de usuario.

Finalmente, cuando se haya conseguido lo anterior, cargar el acumulador con 7 y ejecutar el programa. Análogamente con &C.

Los resultados muestran que &BB5A no sólo escribe en pantalla los caracteres alfanuméricos (letras y números), sino que también obedece los llamados caracteres de control, aunque en el monitor no aparezca nada (7 y &C hacen sonar un «bip» y borrar la pantalla respectivamente).

Esta semana ya sabemos algo más:

- usar el registro acumulador
- llamar a subrutinas del firmware
- predecir el resultado de esas llamadas, en función del contenido del acumulador
- manejar el cargador hexadecimal (HEXER)

Es el momento de experimentar con todo lo que se ha aprendido, tratando de ir más allá de nuestros ejemplos.

La próxima semana, más acerca del acumulador y sus **«parientes»**.

```

10 REM      Hexer
20 REM      Mike Bibby
30 REM (c) AMSTRAD
40 REM      SEMANAL
50 CLEAR
60 ON ERROR GOTO 10
70 size = HIMEM
80 MEMORY &2FF8
90 WHILE -1
100 PRINT: PRINT "Opciones:-":PRINT
110 PRINT "1. Introducir codigo"
120 PRINT "2. Examinar codigo"
130 PRINT "3. Alterar codigo"
140 PRINT "4. Ejecutar codigo"
150 PRINT "5. Salir del programa":PRINT
160 b$ = INKEY$:IF b$="" GOTO 160
170 IF INSTR("12345",b$)=0 GOTO 160
180 b = VAL(b$) : ON b GOSUB 210,400,550
,350,200
190 WEND
200 END
210 INPUT "Direccion de comienzo"; start
$ 
220 IF start$="" THEN start$="3000":PRINT
T:PRINT"Direccion de comienzo = &3000"
230 start = VAL("+"&start$)
240 code$="":PRINT
250 WHILE code$ <> "S" AND code$ <> "s"
260 INPUT "byte"; code$
270 IF code$="" THEN PRINT CHR$(11);: GO
TO 260
280 IF code$="S" OR code$="s" THEN GOTO
330
290 code$ = "&"&code$
300 code = VAL(code$)
310 POKE start, code
320 start = start + 1
330 WEND
340 RETURN
350 INPUT "Direccion de comienzo"; start
$ 
360 IF start$ = "" THEN start$ = "3000"
370 start = VAL("+"&start$)

```

Código máquina

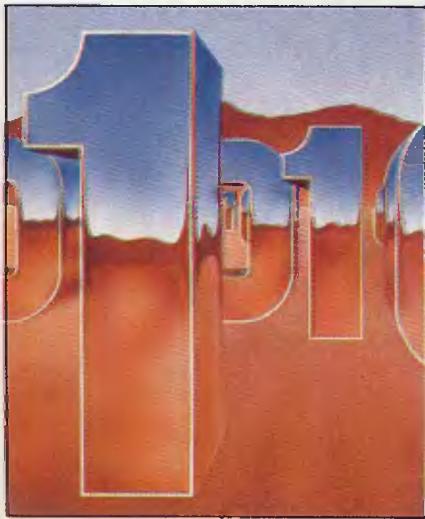
```

380 CALL start
390 RETURN
400 INPUT "Direccion de comienzo"; start
$ 
410 IF start$ = "" THEN start$ = "2FF8"
420 start = VAL("+"&start$)
430 A$ = CHR$(32): PRINT
440 WHILE A$<>"s" AND A$<>"S"
450 PRINT RIGHT$(" "+HEX$(start),4);
"
460 FOR loop = 0 TO 7
470 code$= " "+HEX$(PEEK( start+loop))
)
480 PRINT RIGHT$(code$,3);
490 NEXT
500 PRINT
510 A$=INKEY$ : IF A$="" GOTO 510
520 start = start + 8
530 WEND
540 RETURN
550 INPUT "Alterar desde"; start$ 
560 IF start$ = "" THEN start$ = "3000"
570 start = VAL("+"&start$)
580 code$ = "":PRINT
590 WHILE code$ <> "S" AND code$ <> "s"
600 PRINT RIGHT$(" "+HEX$(start),4);
";RIGHT$(" "+HEX$(PEEK(start)),2)); "
;
610 INPUT code$ 
620 IF code$="S" OR code$="s" THEN GOTO
670
630 IF code$ = "" THEN GOTO 660
640 code$ = "&"&code$ 
650 POKE start,VAL(code$)
660 start = start + 1
670 WEND
680 RETURN

```



P
ara que tus dedos
no realicen el trabajo duro, M.H. AMS-
TRAD lo hace por ti. Todos los listados que incluyen
este logotipo se encuentran a tu disposición en un cas-
sette mensual, solicítanoslo.



MAD ADDER

*No hay nada como el hambre para resolver laberintos donde se encuentra la comida que necesitamos para sobrevivir.
Vamos, pues, a guiar a nuestro simpático protagonista: ¡a ver cuánto somos capaces de hacerlo crecer!*

La pantalla está dividida en dos áreas distintas, una dentro de otra, a las cuales se accede a través de unas puertas situadas en los laterales.

El choque con las paredes del laberinto es mortal, y cualquier serpiente que, al retroceder, marche sobre sus propios pasos, resultará destruida sin misericordia.

A medida que la serpiente come, lógicamente engorda y crece, con lo cual arrastramos por la pantalla una «cola» cada vez más larga. Eso dificulta considerablemente nuestros movimientos y exige gran concentración y habilidad.

Entre tanto, la comida se deteriora con el tiempo, y se convierte en otro obstáculo que habrá que superar.

La serpiente se controla con las teclas A,Z, < y > o el joystick. Si estos mandos no convencen, basta con modificar los números entre paréntesis de las líneas 27-30.

Las sentencias REM del programa se incluyen por claridad. Pueden ser omitidas sin ningún problema.

VARIABLES PRINCIPALES

NOMBRE FUNCION

SP\$	Almacena el carácter asociado con cada posición de pantalla.
H\$(P)	Cabeza de la serpiente, dependiendo de la dirección hacia la que se está moviendo.
F\$(N)	Comida.
X,Y	Coordenadas de pantalla.
BX,BY	Coordenadas de cada «celdilla» que forma el cuerpo de la serpiente.
DX,DY	Coordenadas de los obstáculos adicionales.
L	Longitud de la serpiente.
SC	Puntuación.
HS	Máxima puntuación.
I,J,K	Variables de uso general.

LINEAS PRINCIPALES DEL PROGRAMA

NUMERO DE LINEA

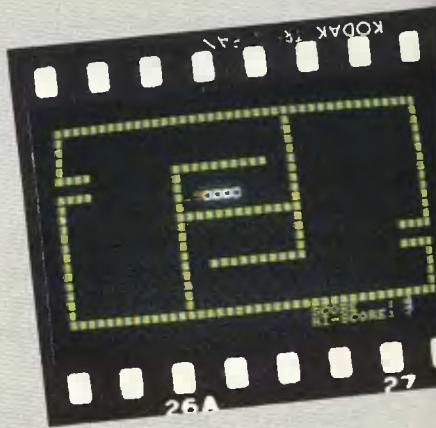
FUNCION

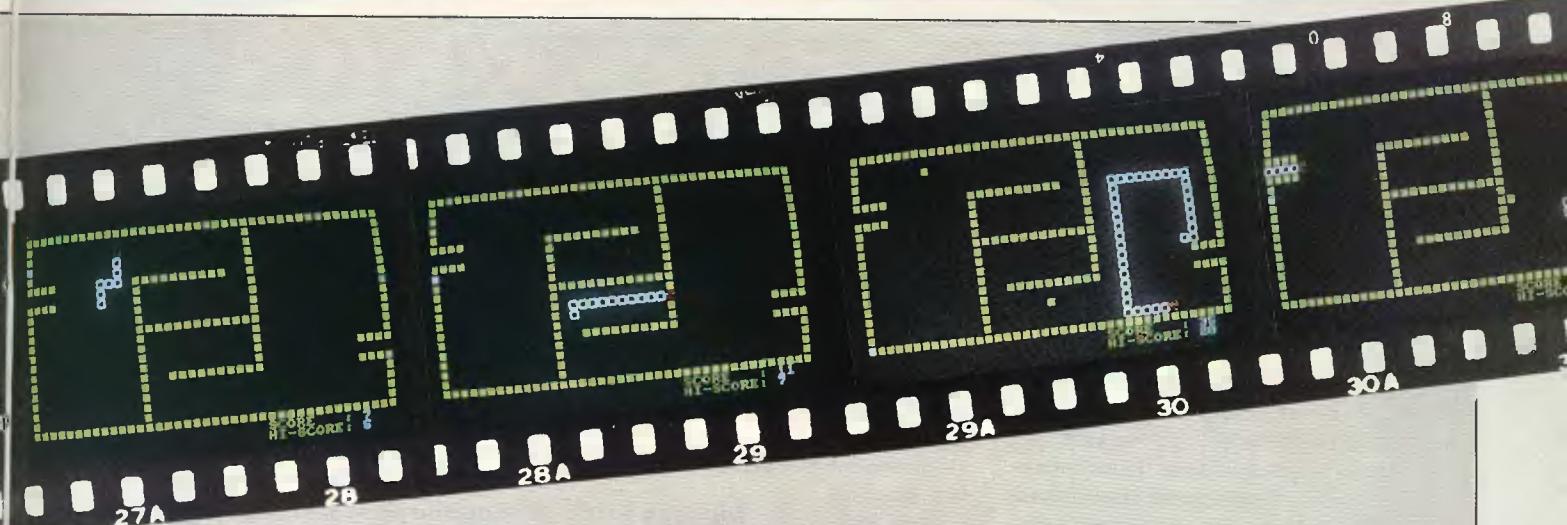
19	Muestra la cabeza de la serpiente y habilita interrupciones para disminuir la cantidad de comida si es necesario.
20	Mira a ver si el alimento ha sido comido.
21	Comprueba choques con las paredes.
22	Borra la cabeza
23	Si no hay cuerpo de la serpiente que «pintar», borra la variable que contiene la posición de la cabeza y salta a la línea 27.
24	Pinta la nueva «celdilla» del cuerpo.
25	Almacena la posición de cada «celdilla» del cuerpo de la serpiente, para posteriormente borrarlas.
26	Cuando la serpiente es lo suficientemente larga, salta a la subrutina que borra la «celdilla» final.
27-36	Asigna nueva dirección de movimiento a la serpiente, según se pulse el teclado o se mueva el joystick.
37	Comienza de nuevo.

```

1 REM
2 REM      MAD ADDER
3 REM      by M.Andrews
4 REM      ****
5 REM  (c) AMSTRAD SEMANAL
6 REM      ****
7 REM
8 REM
9 MODE 1:BORDER 10:INK 0,10:INK 1,24:INK
2,26:INK 3,6
10 DIM SP$(40,21),F$(9),H$(4),BX(200),BY
(200),DX(50),DY(50)
11 GOSUB 44:REM Define caracteres
12 GOSUB 57:REM Dibuja la pantalla
13 GOSUB 78:REM Inicialización
14 GOSUB 88:REM Indicador de comida
15 EVERY 150 GOSUB 96:REM Reduce la cantidad de comida
16 REM      ++++++
17 REM      +PROGRAMA PRINCIPAL+
18 REM      ++++++
19 LOCATE X,Y: PEN 3:PRINT H$(P):EI:D1
20 IF SP$(X,Y)=F$(N) THEN GOSUB 109
21 IF SP$(X,Y)=A$ OR SP$(X,Y)=B$ THEN 11
5
22 FOR T=1 TO 100:NEXT:LOCATE X,Y:PRINT
  "
23 IF SC=0 THEN SP$(X,Y)=""":GOTO 27
24 SP$(X,Y)=B$":LOCATE X,Y: PEN 2:PRINT B$
25 BX(I)=X:BY(I)=Y:L=L+1:I=I+1:IF I=200
THEN I=0
26 IF L>SC THEN GOSUB 38
27 IF INKEY(31)=8 OR JOY(0)=8 THEN P=1
28 IF INKEY(39)=8 OR JOY(0)=4 THEN P=2
29 IF INKEY(69)=8 OR JOY(0)=1 THEN P=3
30 IF INKEY(71)=8 OR JOY(0)=2 THEN P=4
31 IF P=1 THEN X=X+1
32 IF P=2 THEN X=X-1
33 IF P=3 THEN Y=Y-1
34 IF P=4 THEN Y=Y+1
35 IF X=0 THEN X=48:Y=15
36 IF X=41 THEN X=1:Y=7
37 GOTO 19
38 SP$(BX(J),BY(J))=""":LOCATE BX(J),BY(J)
):PRINT  "

```





```

39 L=L-1:J=J+1:IF J=200 THEN J=0
40 RETURN
41 REM ++++++*****+*****+*****+
42 REM + DEFINE CARACTERES +
43 REM ++++++*****+*****+*****+
44 SYMBOL 242,&7E,&F7,&FF,&FF,&C0,&F8,&7
F,&3E
45 SYMBOL 243,&7E,&EF,&FF,&FF,&3,&F,&FE,
&7C
46 SYMBOL 244,&72,&F3,&F3,&B3,&F7,&F7,&F
E,&7C
47 SYMBOL 245,&3E,&7F,&EF,&EF,&CD,&CF,&C
F,&4E
48 SYMBOL 246,&7E,&FF,&C3,&C3,&C3,&C3,&F
F,&7E
49 A$=CHR$(233):B$=CHR$(246):C$=CHR$(241
)
50 H$(1)=CHR$(242):H$(2)=CHR$(243)
51 H$(3)=CHR$(244):H$(4)=CHR$(245)
52 F$(1)="1":F$(2)="2":F$(3)="3":F$(4)="
4":F$(5)="5"
53 F$(6)="6":F$(7)="7":F$(8)="8":F$(9)="
9":RETURN
54 REM ++++++*****+*****+*****+
55 REM + DIBUJA PANTALLA +
56 REM ++++++*****+*****+*****+
57 FOR X= 1 TO 40:Y= 1:GOSUB 73:NEXT
58 FOR Y= 2 TO 6:X= 1:GOSUB 73:NEXT
59 FOR X= 2 TO 4:Y= 6:GOSUB 73:NEXT
60 FOR X= 1 TO 4:Y= 8:GOSUB 73:NEXT
61 FOR Y= 9 TO 20:X= 1:GOSUB 73:NEXT
62 FOR Y= 2 TO 14:X=40:GOSUB 73:NEXT
63 FOR X=37 TO 39:Y=14:GOSUB 73:NEXT
64 FOR X=37 TO 39:Y=16:GOSUB 73:NEXT
65 FOR Y=16 TO 20:X=40:GOSUB 73:NEXT
66 FOR X= 1 TO 40:Y=21:GOSUB 73:NEXT
67 FOR X=15 TO 25:Y=11:GOSUB 73:NEXT
68 FOR X=15 TO 23:Y= 6:GOSUB 73:NEXT
69 FOR Y= 6 TO 20:X=14:GOSUB 73:NEXT
70 FOR X=17 TO 25:Y=16:GOSUB 73:NEXT
71 FOR Y= 2 TO 16:X=26:GOSUB 73:NEXT
72 RETURN
73 SP$(X,Y)=A$ :LOCATE X,Y: PEN 1:PRINT a$
74 RETURN
75 REM ++++++*****+*****+*****+
76 REM + INICIALIZA +

```

```

77 REM ++++++*****+*****+*****+
78 PEN 1:IF SC>HS THEN HS=SC
79 LOCATE 27,22:PRINT "SCORE : "
80 LOCATE 27,23:PRINT "HI-SCORE: "
81 LOCATE 36,23: PEN 2:PRINT HS
82 X=33:Y=6:P=INT(4*RND(1)+1)
83 SP$(X,Y)=H$(P):L=0:I=0:J=0:K=0:SC=0
84 RETURN
85 REM ++++++*****+*****+*****+
86 REM + MUESTRA COMIDA +
87 REM ++++++*****+*****+*****+
88 N=INT(7*RND(1)+3)
89 FX=INT(38*RND(1)+2):FY=INT(18*RND(1)+
2)
90 IF SP$(FX,FY)<>"":THEN 89
91 SP$(FX,FY)=F$(N):LOCATE FX,FY: PEN 3:P
RINT F$(N)
92 RETURN
93 REM ++++++*****+*****+*****+
94 REM + REDUCE COMIDA +
95 REM ++++++*****+*****+*****+
96 N=N-1:IF N>0 THEN SOUND 1,500,10,5
97 WHILE N=
98 SP$(FX,FY)=A$ :LOCATE FX,FY: PEN 1:PRIN
T A$
99 SOUND 1,1000,5,7
100 K=K+1:DX(K)=FX:DY(K)=FY:N=INT(7*RND(
1)+3)
101 FX=INT(38*RND(1)+2):FY=INT(18*RND(1)+
2)
102 IF SP$(FX,FY)<>"":THEN 101
103 WEND
104 SP$(FX,FY)=F$(N):LOCATE FX,FY: PEN 3:
PRINT F$(N)
105 RETURN
106 REM ++++++*****+*****+*****+
107 REM + DEVORA +
108 REM ++++++*****+*****+*****+
109 SOUND 1,20,10,5:SP$(FX,FY)=H$(P)
110 SC=SC+N:LOCATE 36,22: PEN 2:PRINT SC
111 GOSUB 88:RETURN
112 REM ++++++*****+*****+*****+
113 REM + CHOQUE +
114 REM ++++++*****+*****+*****+
115 INK 0,7,15:BORDER 7,15
116 FOR T=1 TO 500:SOUND 132,1000,5,5:NE
XT

```

117 INK 0,10:BORDER 10
 118 LOCATE 10,25: PEN 3:PRINT C\$: " PULSA
 ESPACIO ";
 119 REM ++++++*****+*****+*****+
 120 REM + NUEVO JUEGO +
 121 REM ++++++*****+*****+*****+
 122 W\$=INKEY\$: IF W\$<>" " THEN 122
 123 LOCATE 10,25:PRINT SPC(22)
 124 SP\$(FX,FY)="" :LOCATE FX,FY:PRINT " "
 125 IF L>0 THEN GOSUB 129
 126 IF SP\$(X,Y)=A\$ THEN LOCATE X,Y: PEN 1
 :PRINT A\$:PRINT CHR\$(7)
 127 IF K>0 THEN GOSUB 131
 128 FOR T=1 TO 500:NEXT:GOTO 13
 129 FOR R=1 TO L:GOSUB 38:PRINT CHR\$(7)
 130 FOR T=1 TO 40:NEXT T,R:RETURN
 131 FOR R=1 TO K
 132 SP\$(DX(R),DY(R))="" :LOCATE DX(R),DY(
R):PRINT " "
 133 NEXT:RETURN



Para que tus dedos
 no realicen el trabajo duro, M.H. AMS-
 TRAD lo hace por ti. Todos los listados que incluyen
 este logotipo se encuentran a tu disposición en un cas-
 sette mensual, solicítanos.

GANAR 100.000 PESETAS CON MICROHOBBY AMSTRAD SEMANAL

Porque pretendemos que **AMSTRAD SEMANAL** sea también vuestra revista, hemos abierto una sección en la que se publicarán los mejores programas originales recibidos en nuestra redacción. Vosotros seréis los encargados de realizar estas páginas, en las que podréis aportar ideas y programas interesantes para otros lectores.

Las condiciones son sencillas:

- Los programas se enviarán a **AMSTRAD SEMANAL** en una cinta de cassette, sin protección en el software, de forma que sea posible obtener un listado de los mismos.
- Cada programa debe ir acompañado de un texto explicativo en el cual se incluyan:
 - Descripción general del programa.
 - Tabla de subrutinas y variables utilizadas, explicando claramente la función de cada una de ellas.
 - Instrucciones de manejo.

¡ENVIANOS TU PROGRAMA!
a HOBBY PRESS, S. A. La Granja, s/n. Pol. Ind. Alcobendas (Madrid)

— Todos estos datos deberán ir escritos a máquina o con letra clara para mayor comprensión del programa.

— En una sola cinta puede introducirse más de un programa.

— Una vez publicado, **AMSTRAD SEMANAL** abonará al autor del programa de **15.000 a 100.000 pesetas**, en concepto de derechos de autor.

— Los autores de los programas seleccionados para su publicación, recibirán una comunicación escrita de ello en un plazo no superior a dos meses a partir de la fecha en que su programa llegue a nuestra redacción.

— **AMSTRAD SEMANAL** se reserva el derecho de publicación o no del programa.

— Todos los programas recibidos quedarán en poder de **AMSTRAD SEMANAL**.

— Los programas sospechosos de plagio se eliminarán inmediatamente.

Mercado común

Con el objeto de fomentar las relaciones entre los usuarios de **AMSTRAD**, **MERCADO COMUN** te ofrece sus páginas para publicar los pequeños anuncios que relacionados con el ordenador y su mundo se ajusten al formato indicado a continuación.

En **MERCADO COMUN** tienen cabida, anuncios de ventas, compras, clubs de usuarios de **AMSTRAD**, programadores, y en general cualquier clase de anuncio que pueda servir de utilidad a nuestros lectores.

Envíanos tu anuncio mecanografiado a: **HOBBY PRESS, S.A.**
AMSTRAD SEMANAL.

Apartado de correos 54.062
28080 MADRID

¡ABSTENERSE PIRATAS!

MICROHOBBY AMSTRAD SEMANAL

LE OFRECE AHORA SUS PROGRAMAS YA GRABADOS, PARA QUE VD. NO TENGA QUE TECLEARLOS

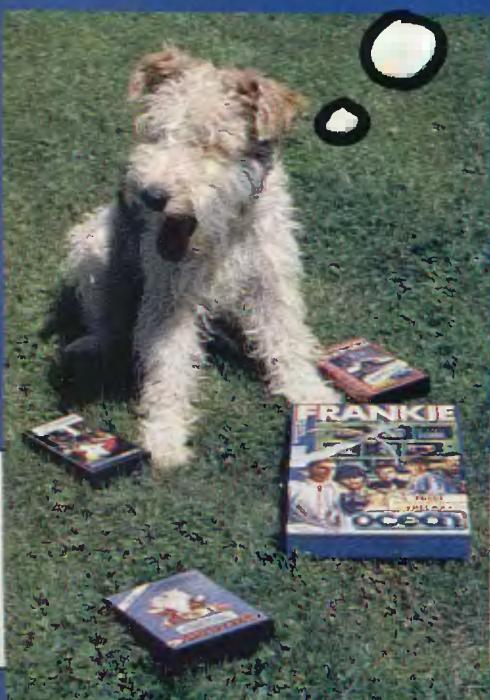
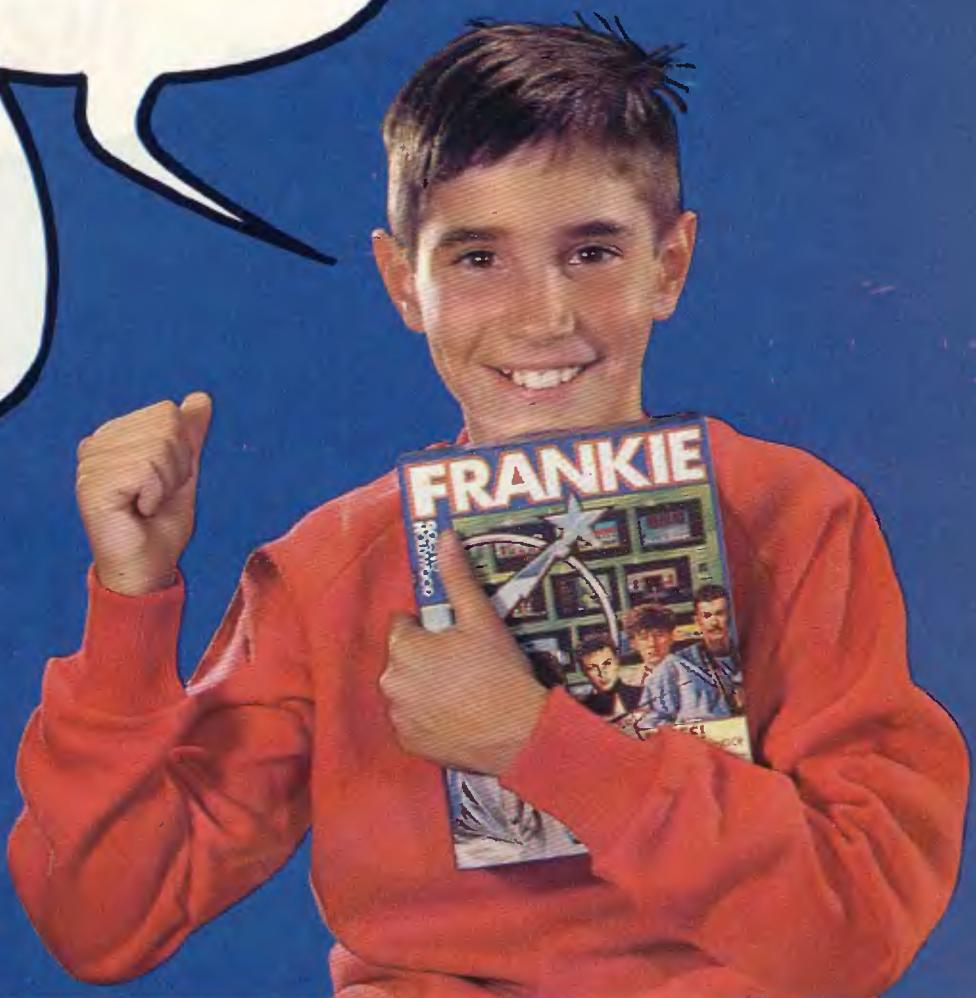
Todos los programadores y aficionados a la microinformática sabemos lo tedioso y propenso a errores que resulta el teclear un listado de un programa. Para facilitar tu labor al máximo y que no tengas que estar horas sobre el teclado de tu ordenador tratando de descifrar incomprensibles mensajes de error, **AMSTRAD SEMANAL** te ofrece cada mes los programas publicados de los cuatro números correspondientes en una cinta de cassette, sólo por **675 ptas.** (*sin más gastos por envío*).

**Envíanos con la menor demora posible,
el cupón correspondiente.**



**SOY SOCIO DEL
CIRCULO DE SOFT
Y CONSIGO
LOS MEJORES PROGRAMAS
A LOS MEJORES
PRECIOS**

**¡Y ANTES
QUE
NADIE!**



circulo de soft
MICROAMIGO S.A.



**¡¡¡SOMOS SOCIOS DE
Y CONSEGUIMOS LOS
A LOS MEJORES**

**¡Y ANTES
QUE
NADIE!**

circulo de soft

CIRCULO DE SOFT LOS MEJORES PROGRAMAS LOS MEJORES PRECIOS!!!



Tú también puedes ser socio de Circulo de Soft. ¡No te cuesta nada!
Envía tu nombre, dirección y teléfono, indicando
la marca, dirección y teléfono, indicando
¡TE SORPRENDEREMOS!

NOMBRE _____
DOMICILIO _____
LOCALIDAD _____
PROVINCIA _____
TEL. _____

ORDENADOR C. P. _____
ORDENADOR _____

MICROAMIGO S.A.
P.º DE LA CASTELLANA, 268
28046-MADRID
TEL.: (91) 733 25 00

GANAR UN AMSTRAD CPC664 PARTICIPANDO EN NUESTRA ENCUESTA

M. H. AMSTRAD, para acercar más y más la revista a los gustos y preferencias de nuestros lectores, plantea la siguiente encuesta que estamos seguros ayudará a hacer una revista abierta a todo tipo de tendencias dentro del mundo de la informática.

Entre todas las cartas recibidas, sorteamos un **AMSTRAD CPC664** y 4 unidades de disco.

Rellenad la encuesta que a continuación os adjuntamos colocando una X en la casilla apropiada y enviadla a:

HOBBY PRESS, S.A.
AMSTRAD SEMANAL

Apartado de Correos 54.062
28080 Madrid

Nombre y apellidos Edad

Domicilio

Localidad

Provincia

Modelo de AMSTRAD

CPC464 CPC664

¿Para qué lo usas?

Juegos Gestión Otros

¿Te interesa la programación?

Sí No

Lenguajes que utilizas

Basic Cód. M. Pascal Logo

¿Te gustaría aprender nuevos lenguajes?

Sí No

¿Cuáles?

Basic Cód. M. Pascal Logo

Forth C

Programas

¿Sueles teclear los programas de las revistas?

Casi todos 1 de 2 1 de 4 1 de 8 Ninguno

¿Qué tipo de programas te interesan?

Juegos Utilidades

Juegos

Marcianos Deportivos Aventuras animadas

Estrategia Inteligencia Aventuras de texto

Utilidades

Procesar textos Bases de datos Hojal calc.

Lenguajes Gestión comercial Gráficos

¿Te gusta que se comenten juegos en tu revista?

Ninguna 1 3 5 Más

¿Cuántas páginas/semana?

Ninguna 1 3 5 Más

¿Qué opinas de los artículos de Cód. M.? ¿Te interesan?

Ninguna 1 3 5 Más

¿Cuántas páginas/semana?

Ninguna 1 3 5 Más

¿Te gustaría que se hablase del ordenador y sus periféricos? ¿Cuántas páginas/semana?

Ninguna 1 3 5 Más

¿Te interesaría una sección dedicada a principiantes en Basic Amstrad? ¿Cuántas páginas/semana?

Ninguna 1 3 5 Más

¿Te interesaría una sección dedicada a gráficos y sonido en el Amstrad? ¿Cuántas páginas/semana?

Ninguna 1 3 5 Más

¿Te gustarían artículos acerca de programas de aplicación comerciales? ¿Cuántas páginas/semana? ... Ninguna 1 3 5 Más

¿Qué secciones añadirías a la revista?

¿Qué secciones quitarías?



Sin duda alguna

A través de esta sección se pretende resolver, en la medida de lo posible, todas las posibles dudas que «atormenten» a todas las personas interesadas en el mundo del AMSTRAD, sean o no poseedores de uno y, si lo son, se encuentren en cualquier nivel de destreza en su manejo.

Semanalmente, aparecen en estas páginas las consultas de la mayor cantidad de usuarios posible; ello redundará en un mejor servicio y en un contacto más estrecho entre todos nosotros a través de la revista.

SIN DUDA ALGUNA está abierta a todos.

Seréis, semana a semana, los encargados de construir esta página con vuestras consultas. En más de una ocasión, aquello que os preocupa ya ha sido contestado antes a otro lector o, por el contrario, puede suceder que determinada consulta aclare muchos quebraderos de cabeza de otros aficionados.

Las cartas «sin duda alguna», nos servirán de gran ayuda. Gracias a ellas podremos ir evaluando vuestras necesidades y, de este modo, modificando el contenido de MICROHOBBY AMSTRAD acorde con ello. ¡Os esperamos!

GAN

M. A
revista
lectores
estamos
abierta
mundo
Entre to
AMSTRAD

Franqueo
Postal

Nombre
Domicilio
Localidad
Provincia

Modelo de
¿Para qué?
¿Te interesan
Lenguajes
¿Te gustaría
¿Cuáles?

Basic Cód. M. Pascal Logo
Forth C

Programas

¿Sueles teclear los programas de las revistas?	Casi todos	<input type="checkbox"/>	1 de 2	<input type="checkbox"/>	1 de 4	<input type="checkbox"/>	1 de 8	<input type="checkbox"/> Ninguno	<input type="checkbox"/>
¿Qué tipo de programas te interesan?	Juegos	<input type="checkbox"/>	Utilidades	<input type="checkbox"/>					
Juegos	Marcianos	<input type="checkbox"/>	Deportivos	<input type="checkbox"/>	Aventuras animadas	<input type="checkbox"/>			
	Estrategia	<input type="checkbox"/>	Inteligencia	<input type="checkbox"/>	Aventuras de texto	<input type="checkbox"/>			
Utilidades	Procesar textos	<input type="checkbox"/>	Bases de datos	<input type="checkbox"/>	Hojal calc.	<input type="checkbox"/>			
	Lenguajes	<input type="checkbox"/>	Gestión comercial	<input type="checkbox"/>	Gráficos	<input type="checkbox"/>			
¿Te gusta que se comenten juegos en tu revista?	Ninguna	<input type="checkbox"/>	1	<input type="checkbox"/>	3	<input type="checkbox"/>	5	<input type="checkbox"/>	Más <input type="checkbox"/>
¿Cuántas páginas/semana?	Ninguna	<input type="checkbox"/>	1	<input type="checkbox"/>	3	<input type="checkbox"/>	5	<input type="checkbox"/>	Más <input type="checkbox"/>
¿Qué opinas de los artículos de Cód. M.? ¿Te interesan?	Ninguna	<input type="checkbox"/>	1	<input type="checkbox"/>	3	<input type="checkbox"/>	5	<input type="checkbox"/>	Más <input type="checkbox"/>
¿Cuántas páginas/semana?	Ninguna	<input type="checkbox"/>	1	<input type="checkbox"/>	3	<input type="checkbox"/>	5	<input type="checkbox"/>	Más <input type="checkbox"/>
¿Te gustaría que se hablase del ordenador y sus periféricos? ¿Cuántas páginas/semana?	Ninguna	<input type="checkbox"/>	1	<input type="checkbox"/>	3	<input type="checkbox"/>	5	<input type="checkbox"/>	Más <input type="checkbox"/>
¿Te interesaría una sección dedicada a principiantes en Basic Amstrad? ¿Cuántas páginas/semana?	Ninguna	<input type="checkbox"/>	1	<input type="checkbox"/>	3	<input type="checkbox"/>	5	<input type="checkbox"/>	Más <input type="checkbox"/>
¿Te interesaría una sección dedicada a gráficos y sonido en el Amstrad? ¿Cuántas páginas/semana?	Ninguna	<input type="checkbox"/>	1	<input type="checkbox"/>	3	<input type="checkbox"/>	5	<input type="checkbox"/>	Más <input type="checkbox"/>
¿Te gustarían artículos acerca de programas de aplicación comerciales? ¿Cuántas páginas/semana?	Ninguna	<input type="checkbox"/>	1	<input type="checkbox"/>	3	<input type="checkbox"/>	5	<input type="checkbox"/>	Más <input type="checkbox"/>
... ¿Qué secciones añadirías a la revista?	Ninguna	<input type="checkbox"/>	1	<input type="checkbox"/>	3	<input type="checkbox"/>	5	<input type="checkbox"/>	Más <input type="checkbox"/>
¿Qué secciones quitarías?									



**Sin
duda
alguna**

A través de esta sección se pretende resolver, en la medida de lo posible, todas las posibles dudas que «atormenten» a todas las personas interesadas en el mundo del AMSTRAD, sean o no poseedores de uno y, si lo son, se encuentren en cualquier nivel de destreza en su manejo.

Semanalmente, aparecen en estas páginas las consultas de la mayor cantidad de usuarios posible; ello redundará en un mejor servicio y en un contacto más estrecho entre todos nosotros a través de la revista.

SIN DUDA ALGUNA está abierta a todos.

Seréis, semana a semana, los encargados de construir esta página con vuestras consultas. En más de una ocasión, aquello que os preocupa ya ha sido contestado antes a otro lector o, por el contrario, puede suceder que determinada consulta aclare muchos quebraderos de cabeza de otros aficionados.

Las cartas «sin duda alguna», nos servirán de gran ayuda. Gracias a ellas podremos ir evaluando vuestras necesidades y, de este modo, modificando el contenido de MICROHOBBY AMSTRAD acorde con ello. ¡Os esperamos!

Corínt por esta linea

¡¡SUSCRIBETE A Microhobby AMSTRAD Y AHORRA 1.600 PTAS!!

SOLICITUD DE CINTAS DE PROGRAMAS Y NUMEROS ATRASADOS

Deseo recibir en mi domicilio, al precio de **675 ptas.** cada una, las siguientes cintas con los programas publicados por **Microhobby AMSTRAD** Marco con una (X) la(s) cinta(s) que deseo:

- Cinta n.º 1 (contiene programas publicados en revistas 1 al 4 inclusive)
- Cinta n.º 2 (contiene programas publicados en revistas 5 al 8 inclusive)
- Cinta n.º 3 (contiene programas publicados en revistas 9 al 12 inclusive)
- Cinta n.º 4 (contiene programas publicados en revistas 13 al 16 inclusive)
- Cinta n.º 5 (contiene programas publicados en revistas 17 al 20 inclusive)

Deseo recibir en mi domicilio, al precio de **150 ptas.** cada uno, los siguientes números atrasados de **Microhobby AMSTRAD**

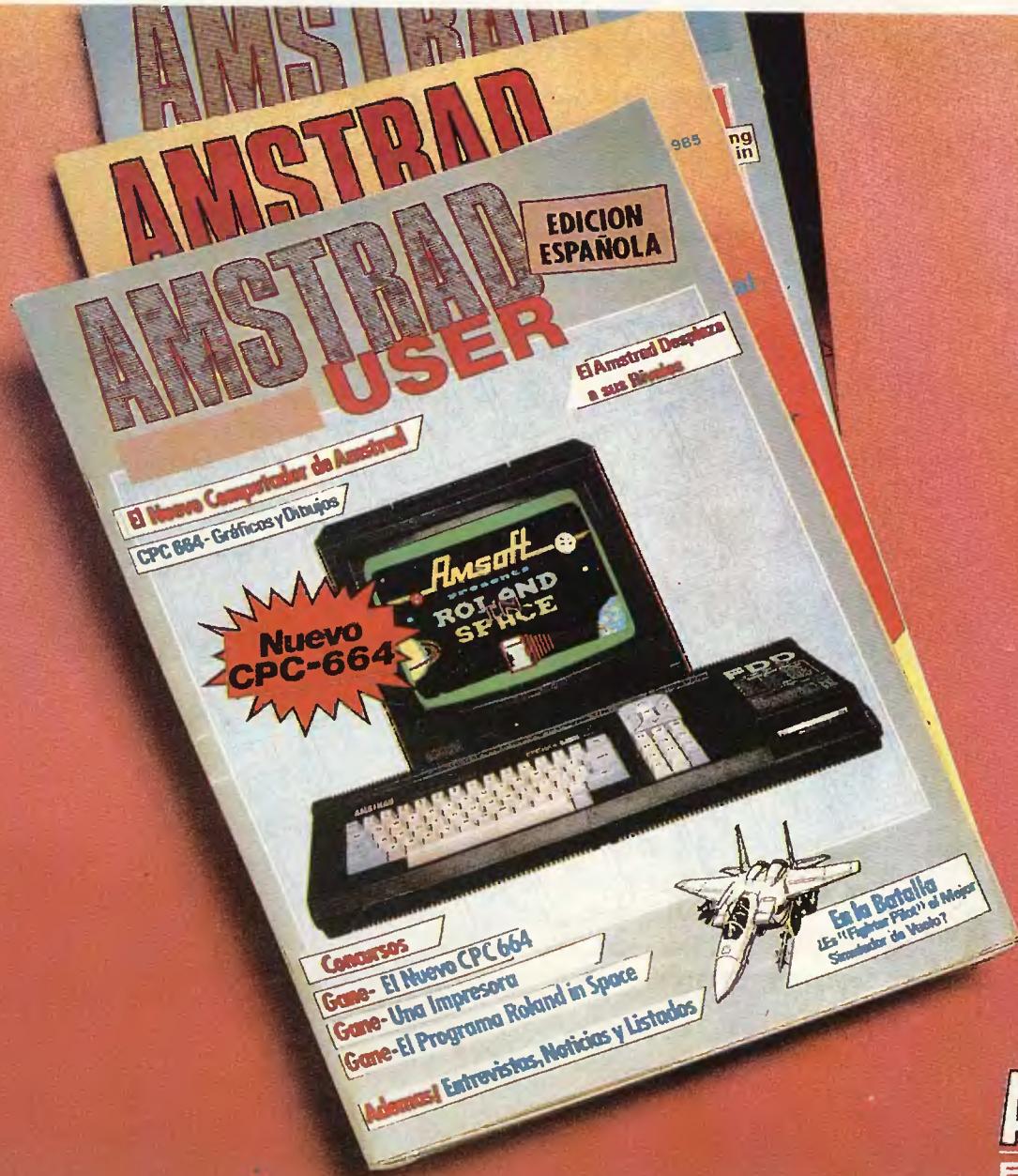
Note: Por razones administrativas, no podemos admitir solicitudes de envío de cintas o números sueltos con pago contra reembolso o Tarjeta de Crédito. Por favor, envíe talón por el importe o giro postal.

Si la forma de pago elegida es talón bancario, remítelo junto con este cupón en un sobre cerrado a la misma dirección. Las cintas de programas se editan una cada mes. Si solicitas varias las recibirás sucesivamente, conforme sean editadas. No se cobran gastos de envío por las cintas y números sueltos.

NOMBRE _____	EDAD _____	
APELLIDOS _____		
DOMICILIO _____		
CIUDAD _____	PROVINCIA _____	
C. POSTAL _____	TELEFONO _____	PROFESION _____
Marco con una (X) en el casillero correspondiente la forma de pago que más me conviene.		
<input type="checkbox"/> Talón bancario adjunto a nombre HOBBY PRESS, S. A. <input type="checkbox"/> Giro Postal N.º _____		
Firma: _____		

**los
ID**

formáti-
vuestro
, repleta
ción, crí-
tcétera.
RAD.



Para más información:
AMSTRAD
ESPAÑA

Dpto. Publicaciones Avd. del Mediterráneo, 9 28007 Madrid

Una Gran Noticia para los Usuarios de AMSTRAD

Apartir del próximo septiembre estará en vuestra tienda de informática, en los quioscos de prensa o —si preferís suscribiros— en vuestro domicilio, la revista **AMSTRAD USER**. Una publicación mensual, repleta de información, con abundantes listados, trucos de programación, crítica de software y periféricos, noticias y novedades, concursos, etcétera.

Para estar al día. Para sacarle aún más partido a tu **AMSTRAD**.



Para más información:
AMSTRAD
ESPAÑA

AMSTRADtm

"Lo increíble" Confirmado por la prensa especializada

tu Micro

Micro, cassette y monitor en plena armonía. Su Basic es el más rápido de su categoría, superando al del Commodore, al del BBC e incluso al del ST.

micro

No hay en el mercado ningún ordenador en este nivel de precio que pueda enfrentarse a él.

Personal Computer World

Su Basic es rápido, más rápido que casi todos los basics de 8 bits y que algunos de 16 bits.

COMPUTER CHOICE

El mejor Basic que he visto.

micros

Calificado de "increíble", las pruebas realizadas así lo han confirmado en casi todos los aspectos... es un equipo con posibilidades fuera de lo común...

Computer Schau

Usuarios y técnicos lo confirman: se ofrece una relación precio/prestaciones que parece imposible.

C'T

Solución total a un precio fenomenal!

SCIENCE VIE MICRO

Se asedia a los distribuidores para conseguir un AMSTRAD. La demanda es desbordante.

Computer persönlich

Por un precio sorprendente se ofrece algo increíble. Un Basic superlativo.

POPULAR Computing WEEKLY

Un ordenador personal extraordinario con unas enormes posibilidades como ordenador de gestión.

micro bit

Su Basic se puede considerar impresionante... tiene unas características no usuales en microordenadores de su categoría.



AMSTRADtm "Lo increíble"